Characterization of anomalous diffusion through convolutional transformers

IOP Publishing Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. **56** (2023) 014001 (20pp) https://doi.org/10.1088/1751-8121/acafb3

Characterization of anomalous diffusion through convolutional transformers

Nicolas Firbas[1], Òscar Garibo-i-Orts[2], Miguel Ángel Garcia-March[3] and J Alberto Conejero[3,*]

[1] DBS—Department of Biological Sciences, National University of Singapore, 16 Science Drive 4, Singapore 117558, Singapore
[2] VRAIN—Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, 46022 València, Spain
[3] IUMPA—Instituto Universitario de Matemática Pura y Aplicada, Universitat Politècnica de València, 46022 València, Spain

E-mail: aconejero@upv.es

Received 13 August 2022; revised 16 December 2022
Accepted for publication 3 January 2023
Published 13 January 2023

Abstract

The results of the Anomalous Diffusion Challenge (AnDi Challenge) (Muñoz-Gil G *et al* 2021 *Nat. Commun.* **12** 6253) have shown that machine learning methods can outperform classical statistical methodology at the characterization of anomalous diffusion in both the inference of the anomalous diffusion exponent α associated with each trajectory (Task 1), and the determination of the underlying diffusive regime which produced such trajectories (Task 2). Furthermore, of the five teams that finished in the top three across both tasks of the AnDi Challenge, three of those teams used recurrent neural networks (RNNs). While RNNs, like the long short-term memory network, are effective at learning long-term dependencies in sequential data, their key disadvantage is that they must be trained sequentially. In order to facilitate training with larger data sets, by training in parallel, we propose a new *transformer* based neural network architecture for the characterization of anomalous diffusion. Our new architecture, the Convolutional Transformer (ConvTransformer) uses a bi-layered convolutional neural network to extract features from our diffusive trajectories that can be thought of as being words in a sentence. These features are then fed to two transformer encoding blocks that perform either regression (Task 1 1D) or classification (Task 2 1D). To our knowledge, this is the first time transformers have been used for characterizing anomalous diffusion. Moreover, this may be the first time that a transformer encoding block has been used with a convolutional neural network and without the need for a transformer decoding block or positional encoding. Apart from being able

* Author to whom any correspondence should be addressed.

to train in parallel, we show that the ConvTransformer is able to outperform the previous state of the art at determining the underlying diffusive regime (Task 2 1D) in short trajectories (length 10–50 steps), which are the most important for experimental researchers.

Keywords: anomalous diffusion, machine learning,
recurrent neural networks, convolutional networks, transformers, attention

(Some figures may appear in colour only in the online journal)

1. Introduction

It could be said that the study of diffusion began in 1827 when Brown first observed the motion, which now carries his namesake, of pollen from *Clarkia pulchella* suspended in water [1]. This movement results from small particles being bombarded by the molecules of the liquid in which they are suspended, as was first conjectured by Einstein and later verified by Perrin [2]. Though Brown never managed to explain the movement he observed, we now know that Brownian motion is a kind of normal diffusion.

To describe diffusion, we can consider the following analogy: let us imagine a particle being an ant, or some other diminutive explorer, we can then think of mean squared displacement (MSD), which can be written as $\langle \mathbf{x}^2 \rangle$, as the portion of the system that it has explored. For normal diffusion such as Brownian motion, the relation between the portion of explored region and time is linear, $\langle \mathbf{x}^2 \rangle \sim t$. As time progresses, the expected value of distance explored by our ant (MSD) will remain constant. In contrast to normal diffusion, anomalous diffusion is characterized by $\langle \mathbf{x}^2 \rangle \sim t^\alpha, \alpha \neq 1$. Anomalous diffusion can be further subdivided into super-diffusion and sub-diffusion, when $\alpha > 1$ or $\alpha < 1$, respectively. To continue using the analogy of our ant, an intuitive example of sub-diffusion would be diffusion on a fractal. In this case, it is easy to see how, as time progresses and our ant ventures into zones of increasing complexity, its movement will in turn be slowed. Thus the relationship of space explored and time will be $\langle \mathbf{x}^2 \rangle \sim t^\alpha, \alpha < 1$. Conversely, if we give our ant wings and have it randomly take flight at random times t_i sampled from $t^{-\sigma-1}$ with flight times positively correlated to the wait time, then for $\sigma \in (0,2)$ we would have a super-diffusive Lévy flight trajectory.

Since the discovery of Brownian motion, many systems have shown diffusive behavior that deviates from the normal one, where MSD scales linearly with time. These systems can range from the atomic scale to complex organisms such as birds. Examples of such diffusive systems include ultra-cold atoms [3], telomeres in the nuclei of cells [4], moisture transport in cement-based materials, the free movement of arthropods [5], and the migration patterns of birds [6]. Anomalous diffusive patterns can even be observed in signals that are not directly related to movement, such as heartbeat intervals and DNA [7, pp 49–89]. The interdisciplinary scope of anomalous diffusion highlights the need for modeling frameworks that are able to quickly and accurately characterize diffusion in real-life scenarios, where data is often limited and noisy.

Despite the importance of anomalous diffusion in many fields of study [8], detection and characterization remain difficult to this day. Traditionally, MSD ($\langle \mathbf{x}^2 \rangle \sim t^\alpha$) and its anomalous diffusion exponent α have been used to characterize diffusion. In practice, computation of MSD is often challenging as we often work with a limited number of points in the trajectories, which may be short and/or noisy, highlighting a need for a robust method for real-world conditions. The problem with using α to characterize anomalous diffusion is that trajectories often have the same anomalous diffusion exponent while having different underlying diffusive regimes. An example would be the motion messenger RNA (mRNA) in a living *E. coli* cell.

The individual trajectories of the mRNA share roughly the same α despite their trajectories being quite distinct [9].

Being able to classify trajectories based on their underlying diffusive regime is useful because it can shed light on the underlying behavior of the particles undergoing diffusion. This could be more important for experimental researchers, which may be more concerned with how a particle moves not necessarily how much it has moved. In this vein, the Anomalous Diffusion challenge (AnDi Challenge) organizers identified the following five diffusive models [10]: the continuous-time random walk (CTRW) [11], fractional Brownian motion (FBM) [12], the Lévy Walk (LW) [13], annealed transient motion (ATTM) [14], and scaled Brownian motion (SBM) [15], with which to classify trajectories (see appendix for a brief introduction to all these methods). This information is not meant to supplant traditional MSD-based analysis, rather, it is meant to give us additional information about the underlying stochastic process behind the trajectory. For example, for a particular exponent α, one may not have access to an ensemble of homogeneous trajectories. Moreover, one cannot assure that all measured trajectories have the same behavior and can therefore be associated with the same anomalous exponent α. In these cases, it may be possible to explain the behavior of the diffusing particles by using what we know about five models mentioned above.

The first applications of machine learning (ML) methods to the study of diffusion aimed to discriminate among confined, anomalous, normal qualitatively, and directed motion [16, 17]. These ML models did not extract quantitative information nor determining did they determine the underlying physical model. At first long short-term memory (LSTM) recurrent neural networks [18] were considered for the analysis of anomalous diffusion trajectories from experimental data in [19]. Later, Muñoz-Gil *et al* [20] computed the distances between consecutive positions in raw trajectories and normalized them by dividing by the standard deviation. Then, their cumulative sums fed random forest algorithms that permit to infer the anomalous exponent α and to classify the trajectory in one of these models, CTRW, FBM, or LW. Random forests and gradient boosting methods were already considered for the study of fractional anomalous diffusion of single-particle trajectories in [21, 22].

The results of the AnDi Challenge [23] showed that ML algorithms outperform traditional statistical techniques in the inference of the anomalous diffusion exponent (Task 1) and in the classification of the underlying diffusion model (Task 2), across one, two, and three dimensions. Some of the most successful techniques consisted of: a couple of convolutional layers combined with some bidirectional LSTM layers and a final dense layer [24], two LSTM layers of decreasing size with a final dense layer [25], a WaveNet encoder with LSTM layers [26] , or extension of classical statistical methods by using deep feed-forward neural networks to cluster parameters extracted from the statistical features of individual trajectories [27]. Recently, there has been more interest in model interpretability, and we have seen a new feature engineering methods that use extreme gradient boosting has been used [28] with state-of-the-art results in the classification task.

As we can see, the best performing methods from the AnDi Challenge were either entirely based on LSTMs recurrent neural networks or incorporated them as part of a larger architecture. For many years, LSTM have been one of the most successful techniques in natural language processing (NLP) and time series analysis. As a matter of fact, Google Translate algorithm is a stack of just seven large LSTM layers [29]. However, since the landmark paper *Attention is All You Need* [30], transformers have become the dominant architecture in NLP, where they have surpassed previous models based on convolutions and recurrent neural networks [31]. Inspired by the transformers' success and by drawing a parallel between the sequential nature of language and the diffusion of a single particle, we propose a new

architecture combining convolutional layers with transformers, that we will call a convolutional transformer: the *Convolutional Transformer* (ConvTransformer)

1.1. The convolutional transformer

The ConvTransformers method has been applied to both the inference the anomalous diffusion exponent α (Task 1 1D) and the determination of the underlying diffusion model (Task 2 1D). As the name suggests, the ConvTransformer uses two convolutional layers followed by a transformer encoding block. However, unlike the transformer in [30], our method uses only two transformer encoding blocks in sequence without a transformer decoding block or positional encoding. The convolutional layers behave as an encoder extracting both linear and non-linear features from the trajectory while retaining spatiotemporal awareness eliminating the need for positional encoding. These features are then passed to the transformer encoding layers where attention is performed upon them. The ConvTransformer structure can be intuitively understood if we consider a single trajectory is akin to a sentence. In this analogy the CNNs are used to create pseudo-words, which are the features produced by the CNNs. Finally, we perform attention twice on the pseudo-words with our transformer encoder, which allows us to determine which features are the most important, and from there we are able to obtain either our α or the underlying diffusive regime model.

The ConvTransformer does not require positional encoding because the CNN kernel moves across the trajectory to create the features. As the CNN kernel moves along the trajectory, it learns positional information, negating the need for positional encoding prior to the transformer encoding block. This was assessed by testing the ConvTransformer on the Task 2 1D with five-fold validation on a training set of size 50 K (32 K for training, 8 K for validation, and 10 K for testing), using the same set of hyper-parameters, with and without the trigonometric encoding scheme used in Vaswani *et al* [30]. A five-fold validation of models trained with and without positional encoding showed showed that model classification accuracy decreased with positional encoding to a mean 72.39% and standard deviation of 4.86 from 75.66% and standard deviation of 1.54. Thus, positional encoding did not improve ConvTransformer performance, and it was omitted from the model.

In figure 1, we show a diagram detailing the structure of the ConvTransformer. As was previously said, the ConvTransformer uses two convolutional layers, one which scales our trajectory up to 20 features, and a second one takes those 20 features and outputs 64 features. This structure allows the CNN to learn lower-level features first and then refine those features in the subsequent layer. Both convolutional layers use a kernel size of 3, a stride of 1, and are followed by a rectified linear unit function (ReLU), and a dropout with a 5% probability of setting a learned parameter to 0 to avoid over-fitting. At the end of the convolutional block, we do a pooling with kernel size 2, which cuts the length of our output in half. This helps conserve video memory (VRAM), optimizing resource consumption and democratizing the model as it can run in consumer-grade hardware. The transformer encoding block follows the basic structure of the transformer encoding block from [30]. It uses wide multi-headed attention with 16 attention heads. The attention mechanism is then followed by layer normalization and dropout. This output feeds two linear layers separated by a ReLU, which ultimately goes to another dropout. This transformer encoding block then feeds into another transformer encoding block. This output of this final transformer encoding block then goes to a max function, which gets the largest value of the output tensor by column. Finally, the output of the ConvTransformer feeds a linear layer that outputs size one for Task 1 1D or size five for each of the categories in Task 2 1D.

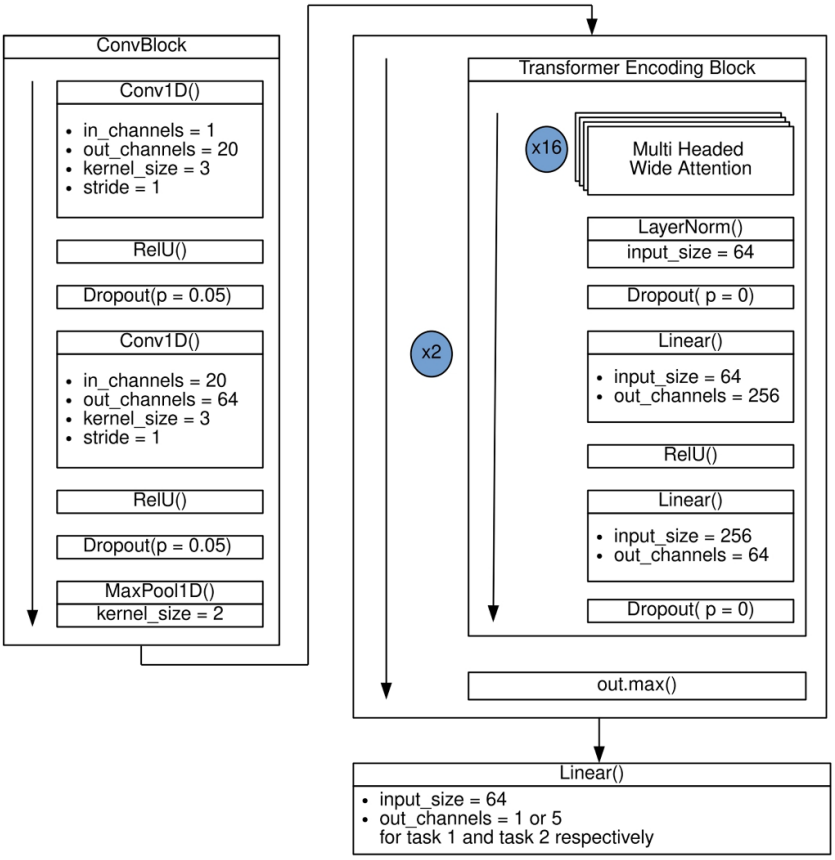

Figure 1. Visual representation of the ConvTransformer structure.

2. Methods

2.1. Generation of training data sets

All of the data sets used to train and test our models were generated with the Python 3 package provided by the AnDi Challenge [23]. The code was made freely available by the AnDi Challenge organizers at: https://zenodo.org/record/4775311#.Ygzvrd-ZOBI.

For the purposes of this research we chose to analyze and generate only one dimensional diffusive trajectories. This is because Vaswani *et al*'s Transformer encoding block [30] was designed for NLP where the inputs to the Transformer would typically be the tokenized words from a sentence. Therefore, each input token, upon which attention is performed, is significantly shorter than our trajectories. Consider that the average length of words in the preceding sentence is about 6, whereas our diffusive trajectories can be hundreds of units long. This results in the features outputted by the CNN layers, which serve as input tokens for out transformer, potentially being hundreds of units in lengths as well.

It could be possible to extend our methodology to higher dimensional trajectories by performing a simple end-to-end concatenation of each of the dimensions of the trajectory or by runing the model on each dimension individually and averaging the results. However, this

Table 1. Table of final hyper-parameters used to train the models for Task 1 and 2 in 1D.

Parameter	Value
Batch size	32
Num. heads	16
CNN dropout	0.05
Trans. dropout	0
Learn rate	0.000 2133
Num. epoch	100
Patience	10

would greatly increase the memory consumption, and run time of our ConvTransformer. The attention mechanism in a Transformer relies on many scalar products performed in parallel, making memory scale non-linearly with input token size. Part of the novelty of this work lies in showing that the attention mechanism in transformers is viable for input tokens, which are greatly longer than those found in NLP, and can therefore be used for characterization of anomalous diffusion without the need to break trajectories into smaller tokens. This allows future work to concentrate on what features of a trajectory each attention head in the transformer encoder is attending to, something that has been previously done in NLP [32]. This should allow researchers to recover some of the model interpret-ability that is lost by using ML methodologies over traditional statistical methods.

Thus, we did not feel that extending our methodology to higher dimensions, using end-to-end dimensional concatenation, provides meaningful additional information about the viability of the attention mechanism for characterization of diffusive trajectories. However, since the completion of our work there has been increased interest in higher dimensional transformers particularly in the form of vision transformers [33].

2.2. Hyper-parameter selection

In order to select the hyper-parameters, we chose a relatively small dataset to train all permutations of these hyper-parameters using five-fold validation to ensure that model performance for a given hyper-parameter set was reliable across runs. The hyper-parameters sets were assessed using a data set with 50 K trajectories of lengths [10, 1000] across both Task 1 1D and Task 2 1D, which was broken up into 32 K training, 8 K validation, and 10 K testing. Then the sets of tested hyper-parameters were evaluated as per the five-fold validation, and the hyper-parameters were chosen for the final model, except the learning rate (LR), which has to be scaled up with respect to training set size and batch size [34]. In the selection process, model performance and feasibility were assessed to ensure that model training could take place on our hardware within a reasonable amount of time. Different sets of hyper-parameters were tested for both Task 1 1D and Task 2 1D but, in the end, we found that the same hyper-parameters work well for both tasks. These final hyper-parameters can be found in table 1.

In order to generalize LR to larger training data sets, we used the results from [34] to relate the noise scale (g) during training to batch size (B), training size (N), and LR (ϵ), as shown in equation (1).

$$g \approx \epsilon \cdot \frac{N}{B} \tag{1}$$

During the training process, we found that an LR of 0.01 worked well across both tasks. Using equation (1), this would give us an equivalent LR of 2.133×10^{-5} when training with

1.35×10^6 trajectories. We used this value as a baseline and we ended up setting an LR of 2.133×10^{-4} for training the final model on a larger data set, as can be seen in table 1.

Our testing revealed that using smaller batches and more heads improved performance. However, decreasing the batch size greatly increases the running time. Ideally, we would have used 32 or more heads. However, we were constrained by our equipment's 8GB video memory (NVIDIA GTX 1070). Thus, with the above set of hyper-parameters, we strove to attain a good balance of speed and performance with our hardware constraints.

2.3. Model training

All model training was conducted in Python 3.8.5 using Pytorch. For simplicity, we trained all our models for Task 1 1D and Task 2 1D on data sets of size 2 million using the same split as before: 75% of the data for training and 25% for testing, with the training set further broken as 90% for training and 10% for validation to be used by Early Stopping to halt the training [35]. Thus, the final training sets were broken up into: 1.35×10^6 million trajectories for training, 1.50×10^5 for validation, and 5×10^5 for testing.

For Task 1 1D, we trained 12 models, each model corresponding to a batch of trajectory lengths: $[10, 20]$, $[21, 30]$, $[31, 40]$, $[41, 50]$, $[51, 100]$, $[101, 200]$, $[201, 300]$, $[301, 400]$, $[401, 500]$, $[501, 600]$, $[601, 800]$, and $[801, 1000]$. All datasets are of size 2×10^6, with the aforementioned training/test/validation split ratio. By default, the andi-datasets package [36] generates trajectories with anomalous exponent $\alpha \in [0.05, 2)$ in intervals of 0.05. This means that 39 different alpha values can be generated, and there are five diffusion models for a total of 195 different kinds of trajectories. After generating data sets of size 2×10^6 would ensure that each of the 195 combinations of diffusion model and α has a representative sample size of about 10^5. Naturally, this is lower after splitting the data sets into training, test, and validation. However, data sets of this size were a good compromise between model performance and training time on our hardware.

In order to accelerate the training of the 12 models, we reduced the patience of our early stopping function from ten, used in hyper-parameter selection, to five while maintaining the number of epochs at 100. Additionally, we conducted the training so that models inherit the parameter state of a previously trained model. This has two advantages: firstly, it indirectly exposes the model to more unique trajectories, as the model will inherit a parameter state that was trained on a different data set, thus reducing overfitting. Secondly, it jump-starts the training of each model with a parameter state that was trained on longer trajectories, which should contain relevant information for classifying shorter trajectories.

To implement this training scheme, we trained the first ConvTransformer on the *easiest* dataset, trajectories of length $[801-1000]$, as can be inferred from our testing and the results in the AnDi Challenge [23]. Then the parameter state of this model is used as the starting parameters state for the next model, which will be trained on trajectories of lengths $[601-800]$ and so forth until the final model is trained on trajectories of length $[10, 20]$. Once we have completed the first training pass through, we loop back to the top and repeat the process, with model $[801, 1000]$ from round two inheriting the parameter state of model $[10, 20]$ from the first training round. Finally, the round two models are tested on every testing data set, and the best models at each trajectory length are selected. This final selection process resulted in 11 models as the models trained on $[401, 500]$, $[601-800]$, and $[501-600]$ outperformed other models at their native trained trajectory lengths. Thus, our compiled model for Task 1 1D consists of 11 models, each of which is in charge of certain trajectory lengths.

Finally, for Task 2 1D, a single model was used across all trajectory lengths (10–1000) as we were to improve upon the state art while maintaining parsimony, as we show in figure 3.

The transformer was trained using 100 epochs, patience of 10, and a single data set with trajectory lengths $[10, 1000]$.

3. Results

We use the AnDi Interactive Tool[4] extensively in our testing in order to be able to assess our model's performance against the current state of the art. The AnDi interactive tool relies on the AnDi test data set. The one dimensional AnDi test data set is composed of of 10 K synthetic trajectories trajectories of lengths $[10-1000]$. Each trajectory in the data set belongs to one of the five underlying diffusive regimes (ATTM, CTRW, LW, FBM, SBM) and has an anomalous diffusion exponent $\alpha \in [0.05, 2]$. In Addition the trajectories are corrupted at three different noise levels $\sigma_{\text{noise}} \in \{0.1, 0.05, 1\}$ and the true α values and true underling diffusive regime of each trajectory is kept hidden to avoid having competitors fit their models to this specific data set [23]. Thus, to gain further insight into our model's performance under different combinations of trajectory type (ATTM, CTRW, LW, FBM, SBM), trajectory length, anomalous diffusion exponent (α), and signal to noise ratio (SNR), defined as SNR$= \sigma_{\text{disp}}/\sigma_{\text{noise}}$, where σ_{disp} is the standard deviation of the displacements and σ_{noise} is the standard deviation of the Gaussian white noise. We generated data sets for all of the permutations seen in table 2.

We have used the performance of our model on these data sets to make the figures in the following sections. In order to improve model comparability we will use the following metrics of performance:

- The mean average error (MAE) is defined as

$$\frac{1}{N}\sum_{j=1}^{N} |\alpha_{j,\text{pred}} - \alpha_{j,\text{true}}|, \tag{2}$$

  where $\alpha_{\cdot,\text{pred}}$ and $\alpha_{\cdot,\text{true}}$ are the predicted and true α values respectively.
- The F1-Score is the harmonic mean of precision and recall and it is defined as

$$\frac{\text{true pos.}}{\text{true pos.} + \frac{1}{2}(\text{false pos.} + \text{false neg.})}. \tag{3}$$

  For our purposes, we have used the micro averaged F1-Score, that is biased by class frequencies, as it has been considered in the AnDi Challenge.

Using the AnDi Challenge interactive tool, we can see how the ConvTransformer would have performed in the AnDi Challenge in table 3. Overall the ConvTransformer would have placed in the middle of the top ten of the AnDi Challenge. However, ConvTransformer shines in classifying short trajectories (Task 2 1D). If we restrict the AnDi testing data set to only trajectories of length $[10, 50]$ we can see that the ConvTrasnsformer would outperform the three best models in the Andi Challenge (3). Though it should be noted that, for Task 2 1D, ConvTransformer performance is within margin of error of team UPV-MAT [24]. The ConvTransformer was able to achieve this performance while using a smaller training set than team UPV-MAT, 1.35×10^6 trajectories versus 4×10^6 trajectories used by team UPV-MAT.

[4] http://andi-challenge.org/interactive-tool/.

Table 2. A testing dataset of size 2000 was generated for all the permutations of each row in the table.

Diff. model	Traj. length	SNR	α
ATTM	$10, 20, \ldots, 50, 100, 200, \ldots, 600, 800, 1000$	1, 2	$0.1, 0.2 \ldots 1.0$
CTRW	$10, 20, \ldots, 50, 100, 200, \ldots, 600, 800, 1000$	1, 2	$0.1, 0.2 \ldots 1.0$
FBM	$10, 20, \ldots, 50, 100, 200, \ldots, 600, 800, 1000$	1, 2	$0.1, 0.2 \ldots 1.9$
LW	$10, 20, \ldots, 50, 100, 200, \ldots, 600, 800, 1000$	1, 2	$1.0, 1.1 \ldots 1.9$
SBM	$10, 20, \ldots, 50, 100, 200, \ldots, 600, 800, 1000$	1, 2	$0.1, 0.2 \ldots 1.9$

Table 3. Rank is how a model compares to the other models available in the AnDi interactive tool, when the entire 1D AnDi test data set is considered. The MAE, and F1-Scores are calculated on a subset of short trajectories (length $[10, 50]$) of the 1D AnDi test data set. Hence, there can be a discrepancy between models between models that outperform the others in short trajectories, and those that perform relatively better in longer trajectories.

Task	Rank	MAE	Team name	Method
1	7	0.453	ConvTrans.	ConvTransformer
1	1	0.326	UPV-MAT	CNN + biLSTM [24]
1	2	0.329	HNU	LSTM [26]
1	3	0.385	eduN	RNN + Dense NN [25]
Task	**Rank**	**F1 score**	**Team name**	**Method**
2	6	0.563	ConvTrans.	ConvTransformer
2	1	0.499	eduN	RNN + Dense NN [25]
2	2	0.560	UPV-MAT	CNN + biLSTM [24]
2	3	0.525	FCI	CNN [37, 38]

3.1. Regression of the anomalous diffusion exponent (Task 1 1D)

We first show the performance of our model with the AnDi Interactive tool, see figure 2 .The ConvTransformer, as well as the top performers in the AnDi Challenge seen in table 3, had the most difficulty inferring the α of ATTM and SBM diffusive regimes, with ATTM being far more problematic. This makes sense if we consider the way ATTM trajectories are generated. The displacements of particles undergoing ATTM are distributed $BM(D, t, \Delta t)$, where BM generates a Brownian motion trajectory of length t sampled at times Δt, with diffusivity coefficient D. Additionally, in ATTM D is re-sampled every $t \sim D^{\sigma/\alpha}$. This means that every time t a particle in ATTM will change diffusive regime in a manner that may obscure α.

Similar to ATTM, SBM also experiences changes in D, the diffusivity coefficient. However, in SBM $D(t) = D\psi(t)$ [39]. On the surface it would appear as though a gradual change in diffusivity should not appear to pose as much difficulty as the regime shifts in ATTM. However, it may have contributed to the difficulty of inferring α in SBM trajectories.

ConvTransformer performance scales as expected with trajectory length, see figure 3. That is to say, as trajectory length increases, the model performance also improves. Notably, performance scaled less erratically than in other models, such as the best performing model in Task 1 1D [24]. Interestingly, noise does not affect model accuracy as heavily at shorter trajectory lengths, and the performance difference between trajectories with respect to the SNR appears to stabilize after trajectories of length $\sim$200.

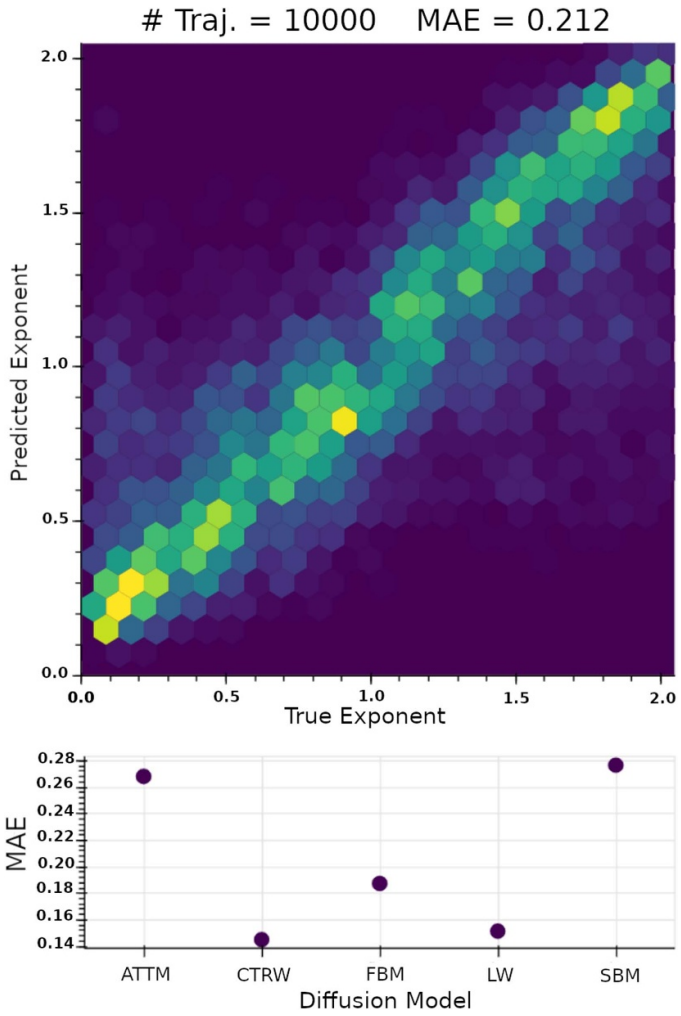

Figure 2. ConvTransformer performance in the regression task, on all 10 K trajectories of the AnDi Test data set, as measured by the AnDi Interactive tool. Lighter colours represent higher frequencies.

In figure 4, we can see the performance breakdown by the underlying diffusive regime. These plots shed more light on the performance issues when regressing α for ATTM. From figure 4, it is evident that most of the difficulty with regressing α in ATTM appears to occur in heavy sub-diffusive trajectories at $\alpha \approx 0.1$, regardless of noise. There, we can see a roughly bi-modal distribution with two clusters at $\alpha \approx 0.4$ and $\alpha \approx 0.9$, with the more significant peak at about 0.9, as shown by the median value line.

Additionally, the ConvTransformer shows similar confusion patterns in the regression task for the SBM model, where it confuses highly super-diffusive trajectories with, roughly, normal diffusion (figure 4). In both of these cases, the regime shift in ATTM and the change in D in SBM could be making heavy anomalous diffusion (both super and sub-diffusion) appear as though it was normal diffusion. However, these effects could also be an artifact of the training data since all diffusive regimes can exhibit normal diffusion, so there will

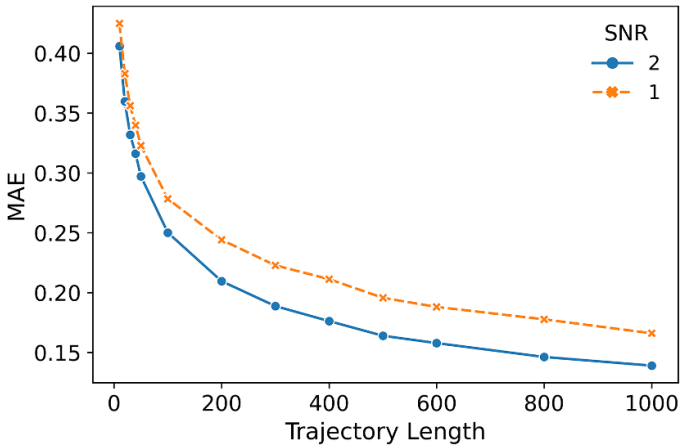

Figure 3. ConvTransformer performance (MAE) in the regression of the anomalous diffusion exponent by SNR as a function of trajectory length.

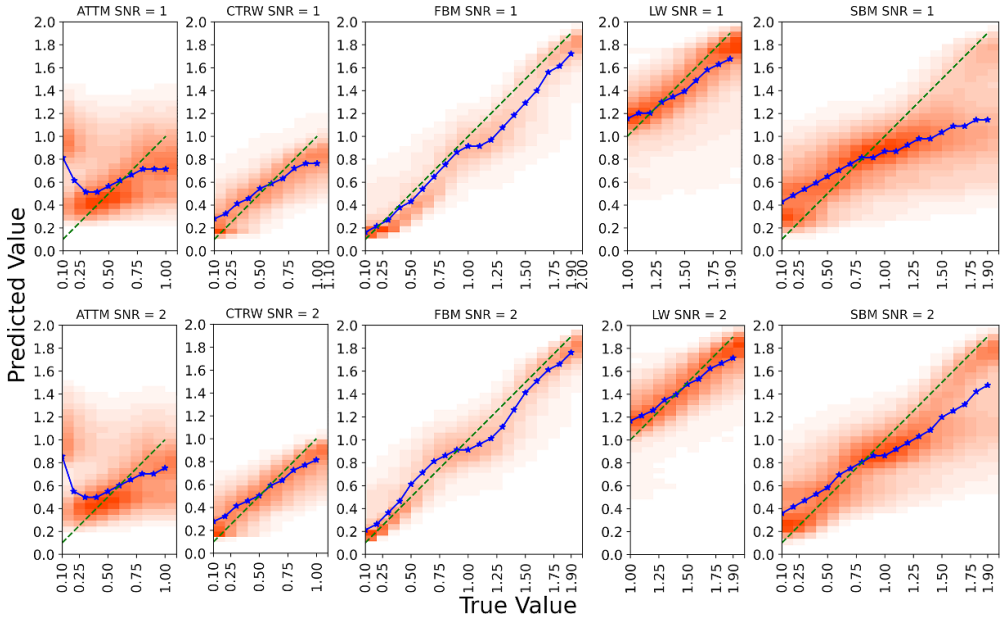

Figure 4. Heat map of ConvTransformer performance in the regression of the anomalous diffusion exponent showing True and Predicted α by the underlying diffusion model. The blue line denotes the median value of the true α values. Predicted values of α are shown from $[0,2]$. However, there were a few instances where the ConvTransformer predicted values marginally less than 0 and greater than 2.

be more trajectories with $\alpha = 1$ than either super or sub-diffusion, or a combination of both effects.

Figure 5 takes a closer look at model performance by trajectory length and type. Once again, most trajectories, with the exception of the ones generated by the SBM model, perform very similarly at SNR 1 and SNR 2, which shows notably worse performance at SNR 1 on all

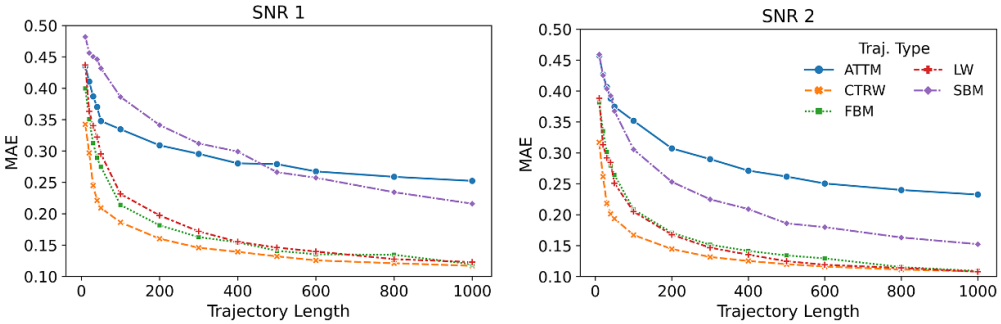

Figure 5. ConvTransformer performance in the regression of the anomalous diffusion exponent (MAE) shown as a function of trajectory length and trajectory type.

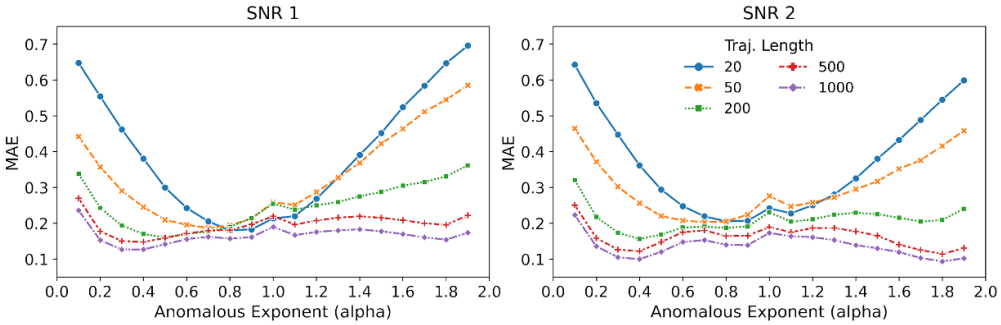

Figure 6. ConvTransformer performance in the regression of the anomalous diffusion exponent (MAE) by trajectory length as a function of α, the anomalous diffusion exponent.

trajectory lengths. It can be verified via the AnDi Interactive Tool that the same effect occurs in the top three models (UPV-MAT, HNU, eduN) shown in table 3, where SBM performance is the most sensitive to additional noise in the trajectory.

When regressing the anomalous diffusion exponent, the sensitivity of ML models to added noise in SBM trajectories warrants further study. Recently, Szarek [40] has encountered a similar lack in resiliency to noise using an RNN-based model, like UPV-Mat, HNU, and eduN models. It appears that the difficulty in working with SBM is an inherent characteristic of SBM trajectories, as opposed to the neural network architecture used for inference of α. This is further substantiated by our transformer-based method encountering the same problem.

In terms of model performance for different values of α, the ConvTransformer perform best at $\alpha \approx .9$ (figure 6). However, for long trajectories, those with 200 or more points, the model performs best roughly between $\alpha \in [0.25, 0.5]$. The latter scenario seems to be closer to the truth if we examine the model performance at various levels of α by trajectory type as in figure 7. Our ConvTransformer performs best roughly in the middle of the domain of α of each trajectory type, with an adequate, though not optimal, performance at $\alpha \approx 1$ across all trajectory types (figure 7). Part of the reason performance is overestimated, at $\alpha \approx 1$, when pooling the trajectory types may be that CTRW and SBM perform best at $\alpha \approx 1$, and these two types of diffusion can be super and sub-diffusive. Thus, they have more testing points and skew the pooled values in figure 6.

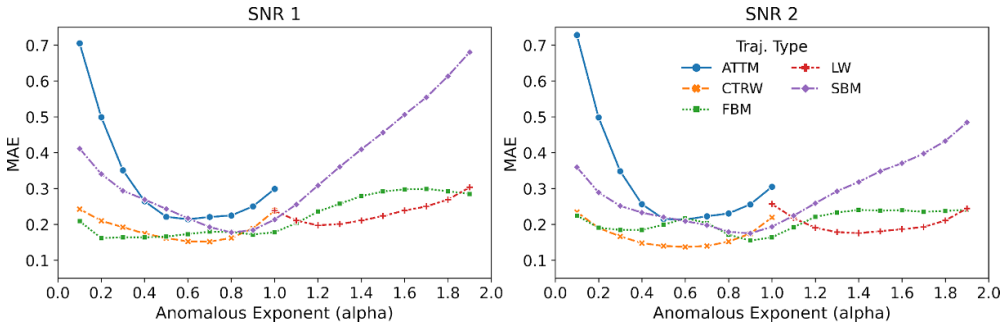

Figure 7. ConvTransformer performance in the regression of the anomalous diffusion exponent (MAE) by underlying diffusive regime as a function of α, the anomalous diffusion exponent.

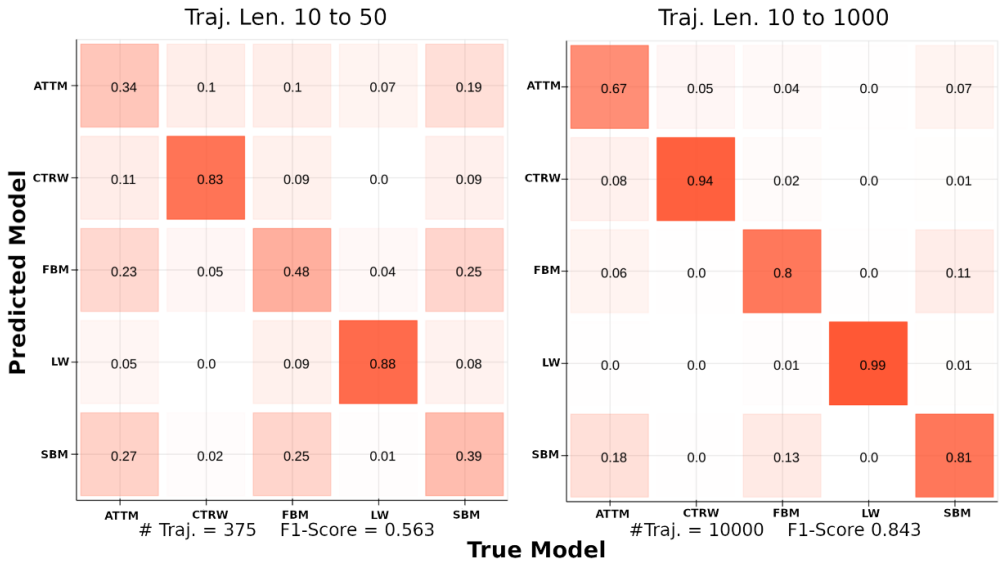

Figure 8. Confusion matrices of ConvTransformer trajectory classification accuracy (Task 2 1D) obtained from the AnDi Interactive Tool.

3.2. Classification of trajectories according to the anomalous diffusion generating model (Task 2 1D)

ConvTransformer performance in the classification of trajectories according to the anomalous diffusion generating model (Task 2 1D) presents results in the average overall, with respect to the ten best models of the AnDi Challenge [23]. However, as we mentioned earlier, our ConvTransformer shines in short trajectories. As with the inference of α (Task 1 1D), ATTM trajectories proved to be the most difficult to work with. These trajectories were most often confused with SBM (figures 8 and 10) , this may be because both models have changes in the diffusivity coefficient D. If we imagine a short ATTM trajectory, where D only changes a few times, the diffusivity coefficient can increase with time ($D \sim \Phi(t)$) in a way that ATTM could mimic SBM.

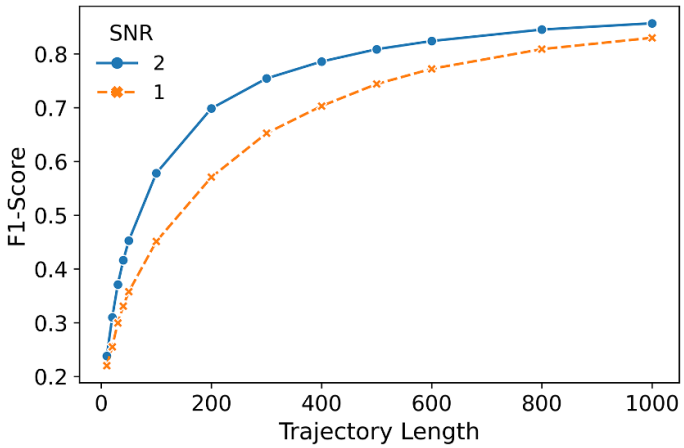

Figure 9. ConvTransformer trajectory classification accuracy (F1-Score) by SNR as a function of trajectory length.

Noise affects ConvTransformer the least at both short and long trajectory lengths (figure 9). Performance at the lower noise data improves faster with respect to trajectory length. The most significant difference between the two curves, in figure 9, occurs at trajectories of length $\sim$200, after which the SNR 1 curve converges towards SNR 2. This indicates that longer trajectories are most helpful when dealing with noisy trajectories that are roughly 200–600 dispersals in length.

When looking at F1-Score by the underlying diffusion model, we can see that ConvTransformer performance varies significantly across our five diffusive regimes (figure 10).

That being said, unlike Task 1 1D, performance change with respect to noise remains fairly constant across the different kinds of diffusion. The outlier to this behavior is short CTRW trajectories at SNR 1 (figure 10). In this case, model performance is better for the shortest trajectories and then drops off before resuming the expected convergence behavior of F1-Score with respect to the trajectory length. The cause of this artifact in the F1-Score is that at SNR 1 and short trajectory lengths ($[10, 50]$), the ConvTransformer is inclined to classify the other diffusive regimes, with the exception of LW, as CTRW (figure 11). It is noteworthy that the ConvTransformer can make the distinction between LW and CTRW as LW can be considered a special case of CTRW [23].

In terms of ConvTransformer performance in classification (Task 2 1D) with regards to α, we can see that ConvTransformer performs better at a value of $\alpha \approx 0.5$ and at the higher-end $\alpha \geqslant 1.5$, with an apparent plateauing behavior at the upper end of the α domain in longer trajectories with lower noise (figure 12). In figure 13 we again look at F1-Score as a function of α. However, this time we look at the relationship in terms of the underlying diffusive model. Most diffusive models retain the relationship seen in figure 12, within their respective domains of α. However, CTRW and LW deviate from this behavior. Both CTRW and LW appear to have a more linear relationship between F1-Score and α, with CTRW performing best at low values of α and LW performing best at higher values of α. This relationship strength (F1-Score $\sim \alpha$) appears to be exacerbated by noise.

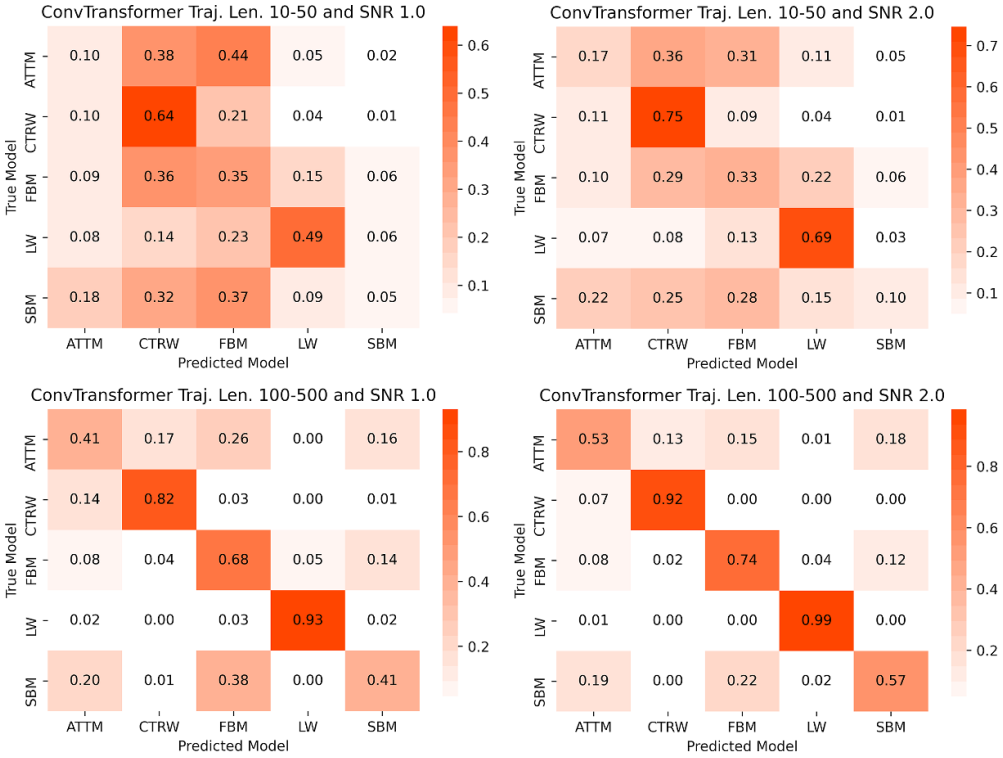

Figure 10. Confusion matrices showing ConvTransformer classification accuracy (Task 2 1D) at different noise levels. Trajectories of length greater than 500 were omitted because although model performance improves at these lengths it does so as we would expect from figure 9 and does not provides further information.

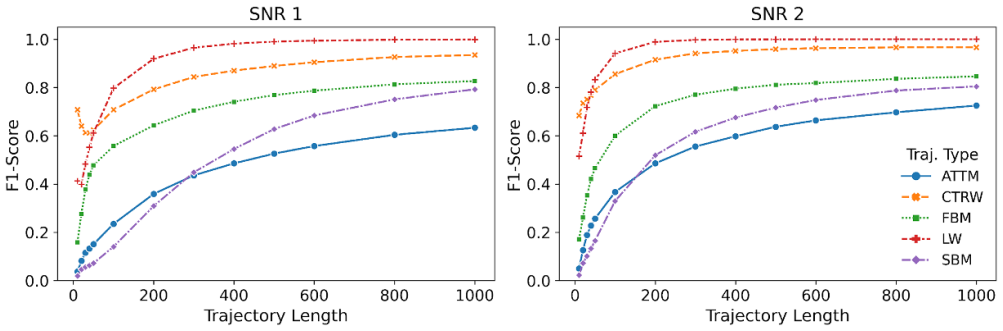

Figure 11. ConvTransformer trajectory classification accuracy (F1-Score) as a function of the trajectory length.

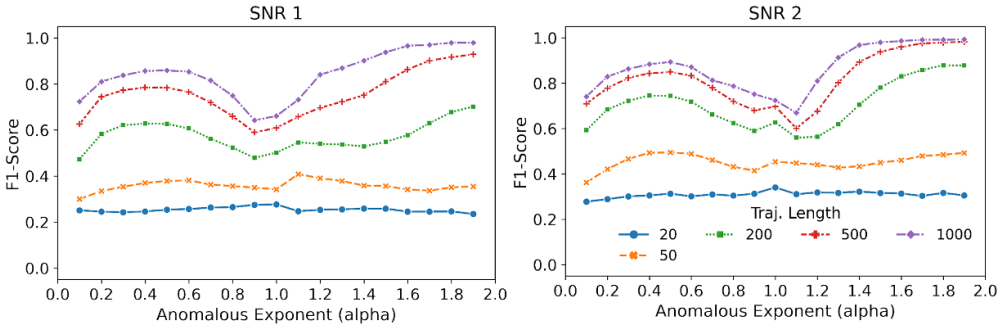

Figure 12. ConvTransformer trajectory classification accuracy (F1-Score) by trajectory length as a function of the anomalous diffusion exponent (α).

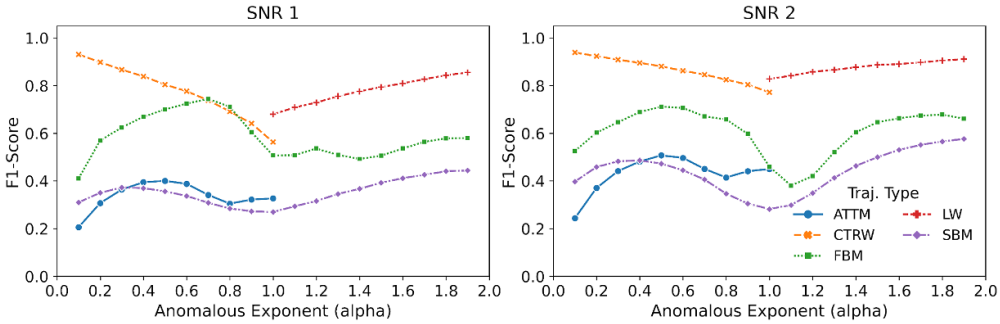

Figure 13. ConvTransformer trajectory classification accuracy (F1-Score) by underlying diffusive regime as a function of the anomalous diffusion exponent (α).

4. Conclusions

The primary purpose of this paper was to introduce our new architecture, the ConvTransformer, for the analysis of anomalous diffusion trajectories. To the best of our knowledge, this is the first transformer based architecture to characterize anomalous diffusion. Indeed it is only recently that anyone else has produced a convolutional transformer (for computer vision) [41, 42], with the development of their models being concurrent with ours. However, our ConvTransformer stands out in that it does not use positional encoding and only uses the transformer encoding block from [30]. As such, it is simpler and easier to implement while still providing state-of-the-art results in trajectory classification (Task 2 1D) in short and noisy trajectories.

Inspired by the success of transformers in NLP we set out to replace the recurrent bidirectional LSTM part of the architecture in [24] by transformers. We have improved the classification of short trajectories accuracy with a model that is trained pretty fast since it can be trained in parallel. When we first started working on this model, there was no native support in PyTorch for transformers. However, when writing this manuscript, transformer encoders and decoders are natively supported. As such, we expect further improvements as ease of implementation and optimized code will lead to more accessibility. This should lead to improved iterations of the model and a finer hyperparametrization tuning. Additionally, the increased optimization and access to newer hardware should increase our ConvTransformers performance for improved usability in experimental research.

Apart from the direct practical implementation of our model in experimental research, going forwards, we would also like to focus on model interpretability. One of the issues plaguing deep learning is the black box effect. When looking at models, we are often only interested in what we can predict or characterize and tend to overlook what we can learn from parameter weighting. Traditionally, parameter weight would allow us to see simple relation between the input features and our desired prediction. For example, birth weight is a strong predictor of adult height [43]. Furthermore, with traditional models like regression, parameter selection leads to discarding information which also informs us about the features that are not relevant to our subject of study. With the rise of deep learning models, we are no longer looking at features, but rather we ingest the data directly and allow our models to discern these features for themselves. With the exception of deep learning models that use feature engineering, as we saw with group UCL and their CONDOR model [27]. The naive approach to modeling brought about by ML means that we not only lose all information about features, but we also do not know what features are important.

As we know from Clark *et al* [32], in the context of NLP, transformer attention heads tend to focus on specific aspects of syntax. For instance, some attention heads may focus entirely on the next token, while others may attend almost entirely to the periods or breaks in a sentence. Following this logic, it is highly likely that some of our ConvTransformer attention heads are specializing on specific features of the trajectories. Hence, a transformer based architecture could be used to determine what trajectory features are important. In this manner, we could recover some model interpretability, and learn from machine learn models in a similar way to how we have traditionally learned from regression.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

N F is supported by the National University of Singapore through the Singapore International Graduate Student Award (SINGA) program. M Á G-M acknowledges funding from the Spanish Ministry of Education and Professional Training (MEFP) through the Beatriz Galindo program 2018 (BEAGAL18/00203) and Spanish Ministry MINECO (FIDEUA PID2019-106901GBI00/10.13039/501100011033). J A C acknowledges funding from Grant PID2021-124618NB-C21 funded by MCIN/AEI/10.13039/501100011033 and by 'ERDF A way of making Europe', by the 'European Union'.

Appendix. Brief description of the theoretical methods

As in the AnDi Challenge, we consider the following five diffusive models: the CTRW [11], FBM [12], the LW [13], ATTM [14], and SBM [15]. These cover the most common diffusive models (CTRW, FBM, LW) along with two more recently introduced models, ATTM and SBM, that explain experiments which are not well described by CTRWs (see e.g. [44]). These two extra models represent an additional challenge because of their similarities to CTRWs.

Generally speaking, in random walks, one can consider the two stochastic variables, step length and waiting time, which may have different probability distributions, to describe motion. CTRWs [11] are a large family of models which arise when the waiting times between

steps are described with a power-law probability distribution, i.e. $\psi(t) \propto t^{-\sigma}$ with $\sigma \geqslant 1$, $t \in [1, \infty)$. Power-law distributions present scale in-variance. Scaling the argument by a constant a introduces a factor $a^{-\sigma}$. Thus all power-laws with a particular scaling exponent are equivalent up to constant factors. Additionally, power-law distributions have a well defined mean for $\sigma > 2$ and finite variance for $\sigma > 3$, which can easily be shown. Here we consider CTRW trajectories with infinite variance in waiting times t ($\sigma \in [1, 3]$), that may or may not have a well defined mean t. It is well known that CTRWs with such behavior show anomalous an diffusing exponent $\alpha = \sigma - 1$. Thus, our CTRW trajectories are sub-diffusive, without a well defined mean t if $\sigma \in (1, 2)$ and are super diffusive with a well defined mean t for $\sigma \in (2, 3]$. Finally, CTRW displacements are sampled from a Gaussian distribution with zero mean and variance determined by diffusion coefficient, D.

LW [13] are a particular case of CTRW where the distribution of displacements is not constant. Displacement times are sampled from $\psi(t) \sim t^{-\sigma-1}$ with $\sigma \geqslant 1, t \in [0, 2)$. The probability displacement length $|\Delta x|$ and wait time are correlated, such that the probability of a moving at time t with displacement $|\Delta x|$ can be given by $\Psi(\Delta x, t) = \frac{1}{2}\delta(|\Delta x| - \nu t)\psi(t)$, where ν is the velocity. In this case the anomalous diffusion coefficient is $\alpha = 2$ if $\sigma \in (0, 1)$ and $3 - \sigma$ if $\sigma \in (1, 2)$.

An ATTM trajectory can be described as the Brownian motion of a particle, where the diffusion coefficient D is a stochastic variable, which varies in time, and is distributed according to a scale-free (Pareto) distribution $P(D) \sim D^{-\gamma}$, $\sigma < \gamma < \sigma + 1$. The stochastic time t at which the particle Brownian motion, with the sampled diffusion coefficient D_t, is extracted from a delta distribution peaked at $D_t^{-\gamma}$. The resulting motion has an anomalous diffusion exponent $\alpha = \sigma/\gamma$.

One can introduce the FBM [12] using a Langevin equation with non-white noise correlated as $\langle \xi(t_0)\xi(t_1)\rangle = 2K_{\mathrm{H}}H(2H-1)|t_0 - t_1|^{2H-2} + 4K_{\mathrm{H}}H|t_0 - t_1|^{2H-2}\delta_{0,1}$ (in one dimension). The FBM shows anomalous diffusion coefficient $1 < \alpha < 2$ if $1/2 < H < 1$ and $0 < \alpha < 1$ if $0 < H < 1/2$ (super-diffusive and sub-diffusive, respectively). For $H = 1/2$ the FBM reproduces Brownian motion.

One can also introduce SBM from a Langevin equation, with Gaussian white noise. However, in this case it has a time dependent diffusivity $K(t)$, which as in ATTM, is related to a power-law. However, SBM trajectories show a power-law dependence with time, and the time related power-law exponent determines the anomalous diffusion coefficient D.

It is important to note that ATTM and CTRW are strictly sub-diffusive, LW is strictly super-diffusive and FBM cannot show $\alpha = 2$ (ballistic behavior). Only SBM and FBM cover the entire range of study from sub to super-diffusion (except at $\alpha = 2$). Finally, while FBM is always ergodic, ATTM, SBM and CTRW are weakly non-ergodic, and LWs are ultra-weakly non-ergodic. All these characteristics, which differentiate the models are expected to be useful for the ML algorithms to differentiate them. Though, ATTM and CTRW are often very similar, thus making it more difficult to differentiate them.

ORCID iDs

Nicolas Firbas https://orcid.org/0000-0003-1724-1807
Òscar Garibo-i-Orts https://orcid.org/0000-0001-8089-1904
Miguel Ángel Garcia-March https://orcid.org/0000-0001-7092-838X
J Alberto Conejero https://orcid.org/0000-0003-3681-7533

References

[1] Brown R 1828 A brief account of microscopical observations made in the months of June, July and August 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies *Phil. Mag.* **4** 161–73
[2] Perrin J 1909 Movement Brownien et realite molec *Ann. Chim. Phys.* **18** 1–114
[3] Sagi Y, Brook M, Almog I and Davidson N 2012 Observation of anomalous diffusion and fractional self-similarity in one dimension *Phys. Rev. Lett.* **108** 093002
[4] Bronstein I, Israel Y, Kepten E, Mai S, Shav-Tal Y, Barkai E and Garini Y 2009 Transient anomalous diffusion of telomeres in the nucleus of mammalian cells *Phys. Rev. Lett.* **103** 018102
[5] Nagaya N, Mizumoto N, Abe M S, Dobata S, Sato R, Fujisawa R and Jing J 2017 Anomalous diffusion on the servosphere: a potential tool for detecting inherent organismal movement patterns *PLoS One* **12** 1–15
[6] Vilk O *et al* 2022 Unravelling the origins of anomalous diffusion: from molecules to migrating storks *Phys. Rev. Res.* **4** 033055
[7] Bunde A 1994 *Fractals in Science with a MS-DOS Program Diskette* (Berlin: Springer)
[8] Manzo C and Garcia-Parajo M F 2015 A review of progress in single particle tracking: from methods to biophysical insights *Rep. Prog. Phys.* **78** 124601
[9] Metzler R, Jeon J-H, Cherstvy A G and Barkai E 2014 Anomalous diffusion models and their properties: non-stationarity, non-ergodicity and ageing at the centenary of single particle tracking *Phys. Chem. Chem. Phys.* **16** 24128–64
[10] Muñoz-Gil G, Volpe G, García-March M A, Metzler R, Lewenstein M and Manzo C 2021 The anomalous diffusion challenge: objective comparison of methods to decode anomalous diffusion *Proc. SPIE* **11804** 1180416
[11] Scher H and Montroll E W 1975 Anomalous transit-time dispersion in amorphous solids *Phys. Rev.* B **12** 2455–77
[12] Mandelbrot B B and Van Ness J W 1968 Fractional Brownian motions, fractional noises and applications *SIAM Rev.* **10** 422–37
[13] Klafter J and Zumofen G 1994 Lévy statistics in a Hamiltonian system *Phys. Rev.* E **49** 4873–7
[14] Massignan P, Manzo C, Torreno-Pina J A, García-Parajo M F, Lewenstein M and Lapeyre G J 2014 Nonergodic subdiffusion from Brownian motion in an inhomogeneous medium *Phys. Rev. Lett.* **112** 150603
[15] Lim S C and Muniandy S V 2002 Self-similar gaussian processes for modeling anomalous diffusion *Phys. Rev.* E **66** 021114
[16] Dosset P, Rassam P, Fernandez L, Espenel C, Rubinstein E, Margeat E and Milhiet P-E 2016 Automatic detection of diffusion modes within biological membranes using back-propagation neural network *BMC Bioinform.* **17** 1–12
[17] Kowalek P, Loch-Olszewska H and Szwabiński J 2019 Classification of diffusion modes in single-particle tracking data: feature-based versus deep-learning approach *Phys. Rev.* E **100** 032410
[18] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
[19] Bo S, Schmidt F, Eichhorn R and Volpe G 2019 Measurement of anomalous diffusion using recurrent neural networks *Phys. Rev.* E **100** 010102
[20] Muñoz-Gil G, Garcia-March M A, Manzo C, Martín-Guerrero J D and Lewenstein M 2020 Single trajectory characterization via machine learning *New J. Phys.* **22** 013010
[21] Janczura J, Kowalek P, Loch-Olszewska H, Szwabiński J and Weron A 2020 Classification of particle trajectories in living cells: machine learning versus statistical testing hypothesis for fractional anomalous diffusion *Phys. Rev.* E **102** 032402
[22] Loch-Olszewska H and Szwabiński J 2020 Impact of feature choice on machine learning classification of fractional anomalous diffusion *Entropy* **22** 1436
[23] Muñoz-Gil G *et al* 2021 Objective comparison of methods to decode anomalous diffusion *Nat. Commun.* **12** 6253
[24] Garibo-i Orts Ó, Baeza-Bosca A, Garcia-March M A and Conejero J A 2021 Efficient recurrent neural network methods for anomalously diffusing single particle short and noisy trajectories *J. Phys. A: Math. Theor.* **54** 504002
[25] Argun A, Volpe G and Bo S 2021 Classification, inference and segmentation of anomalous diffusion with recurrent neural networks *J. Phys. A: Math. Theor.* **54** 294003

[26] Li D, Yao Q and Huang Z 2021 WaveNet-based deep neural networks for the characterization of anomalous diffusion (WADNet) *J. Phys. A: Math. Theor.* **54** 404003
[27] Gentili A and Volpe G 2021 Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR) *J. Phys. A: Math. Theor.* **54** 314003
[28] Kowalek P, Loch-Olszewska H, Łaszczuk Ł, Opała J and Szwabiński J 2022 Boosting the performance of anomalous diffusion classifiers with the proper choice of features *J. Phys. A: Math. Theor.* **55** 244005
[29] Wu Y *et al* 2016 Google's neural machine translation system: bridging the gap between human and machine translation (arXiv:1609.08144)
[30] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need *31st Conf. on Neural Information Processing Systems (NIPS 2017)* (*Long Beach, CA, USA*, *2017*) pp 2–11
[31] Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M and Brew J 2019 Huggingface's transformers: state-of-the-art natural language processing *CoRR*, (arXiv:1910.03771)
[32] Clark K, Khandelwal U, Levy O and Manning C D 2019 What does BERT look at? An analysis of BERT's attention (arXiv:1906.04341)
[33] Khan S, Naseer M, Hayat M, Zamir S W, Khan F S and Shah M 2022 Transformers in vision: a survey *ACM Comput. Surv.* **54** 1–41
[34] Smith S L, Kindermans P J, Ying C and Le Q V 2018 Don't decay the learning rate, increase the batch size
[35] Mehus Sunde B 2020 (available at: https://github.com/Bjarten/early-stopping-pytorch) (Accessed 6 November 2022)
[36] Muñoz-Gil G, Requena B, Volpe G, Garcia-March M A and Manzo C 2021 AnDiChallenge/ANDIs_datasets: challenge 2020 release (v.1.0) *Zenodo* (https://doi.org/10.5281/zenodo.4775311)
[37] Granik N, Weiss L E, Nehme E, Levin M, Chein M, Perlson E, Roichman Y and Shechtman Y 2019 Single-particle diffusion characterization by deep learning *Biophys. J.* **117** 185–92
[38] Bai S, Kolter J Z and Koltun V 2018 An empirical evaluation of generic convolutional and recurrent networks for sequence modeling (arXiv:1803.01271)
[39] dos Santos Maike A F and Menon Junior L 2021 Random diffusivity models for scaled Brownian motion *Chaos Solitons Fractals* **144** 110634
[40] Szarek D 2021 Neural network-based anomalous diffusion parameter estimation approaches for Gaussian processes *Int. J. Adv. Eng. Sci. Appl. Math.* **13** 257–69
[41] Guo J, Han K, Wu H, Xu C, Tang Y, Xu C and Wang Y 2022 CMT: convolutional neural networks meet vision transformers *2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (*New Orleans, LA, USA, 2022*) pp 12165–75
[42] Liu Z, Luo S, Li W, Lu J, Wu Y, Li C and Yang L 2020 Convtransformer: a convolutional transformer network for video frame synthesis *CoRR* (arXiv:2011.10185)
[43] Sørensen H T, Sabroe S, Rothman K J, Gillman M, Steffensen F H, Fischer P and Serensen T I A 1999 Birth weight and length as predictors for adult height *Amer. J. Epidem.* **149** 726–9
[44] Manzo C, Torreno-Pina J A, Massignan P, Lapeyre G J Jr, Lewenstein M and Garc ia García-Parajo M F 2015 Weak ergodicity breaking of receptor motion in living cells stemming from random diffusivity *Phys. Rev.* X **5** 011021