CNN-Transformer with absolute positional encoding optimized for low-dimensional inputs: Applied to estimate sliding drop width

Sajjad Shumaly[1] (✉), Fahimeh Darvish[1], Mahsa Salehi[2], Navid Mohammadi Foumani[2], Oleksandra Kukharenko[1], Hans-Jürgen Butt[1], Ulrich Schwanecke[3], and Rüdiger Berger[1]

[1] Max Planck Institute for Polymer Research (MPI-P), Mainz, Germany
[2] Department of Data Science and Artificial Intelligence, Monash University, Melbourne, VIC, Australia
[3] Department of Computer Science and Media, RheinMain University of Applied Sciences, Wiesbaden, Germany

Abstract. High-speed video recordings are crucial for investigating drop dynamics and their interactions with surfaces. Measuring the width of sliding drops, a key parameter linked to frictional forces, requires additional equipment like cameras or mirrors, complicating experimental setups and limiting observable areas. This study introduces a novel method that simplifies the measurement process by employing artificial neural networks to estimate millimeter-scale drop width directly from side-view video data. Our approach processes raw video footage to dynamically identify features most indicative of drop width. By treating drop behavior as an extrinsic time-series problem, our model effectively captures temporal dependencies in video sequences. We propose a VGG8-inspired architecture optimized for small and low information density video datasets. This architecture is combined with our novel position invariant video processing methodology that efficiently removes non-essential regions, reducing computation time by 84%. We further integrate ConvTran, a state-of-the-art time-series classification model, with an enhanced Absolute Position Encoding, improving the encoding's dot-product and lowering drop width estimation errors. Our novel neural network architecture achieved a root mean square error of 48 µm (1.7 % relative error), where each pixel corresponds to approximately 44 µm. Code and data are open-sourced at: https://github.com/shumaly/position_invariant_cnn_transformer

Keywords: position invariant video processing · low-dimensional absolute positional encoding · extrinsic time series · spatiotemporal CNN–Transformer

1 Introduction

Video analysis of sliding drops enables quantitative studies of sliding forces and liquid–solid interfacial properties [1, 2]. Sliding forces depend on drop width [3, 4]. A recent investigation by Li et al. focused on drops sliding down an inclined

surface, presenting an empirical equation that models the friction force F_f versus drop velocity U [3]:

$$F_f = F_0 + \beta w U \eta \tag{1}$$

Here, β is a dimensionless friction coefficient, w is the width of the drop while sliding, η is the viscosity of the liquid, and F_0 is the friction force extrapolated to velocity $U = 0$. The friction force of drops that just start sliding is described by the Furmidge equation [5–8]:

$$F = k\gamma w(\cos\theta_r - \cos\theta_a) \tag{2}$$

where γ is the liquid–air surface tension, θ_a is the advancing contact angle, θ_r is the receding contact angle, and k is a geometry factor [4, 9]. The Furmidge equation also appears to capture frictional forces at low velocities [10]. The dynamic contact angles vary with velocity and can be easily measured from a side view.

Friction force is essential for detecting surface inhomogeneities, assessing interfacial stability, monitoring viscoelastic energy dissipation [43], and also is critical in anti-icing [41] and surface coating quality [42]. However, determining the drop width during a standard sliding drop experiment remains a challenging task. Adding cameras for bottom- or top-view measurements is not feasible since these views show the drop's central width, not the drop's contact line width. The drop's contact line width is narrower on surfaces with contact angles $> 90°$. Front-view imaging of drops is feasible by installing two mirrors or a second, time-synchronized high-speed camera [10, 11]. However, it is limited to a sliding length of only ≈ 1.5 cm, as the drop moves toward the mirror and cannot stay within the camera's focus range for an extended period. To address these limitations, Shumaly et al. recently proposed a deep learning-based multivariate time-series analysis approach that leverages side-view measurements to estimate the front-view drop width, eliminating the need to add additional cameras or devices and without limiting sliding length [12].

Practical significance Previous research has relied on predefined measures extracted from side-view videos—such as contact angles, drop length, height, and the velocity of the drop's center—to estimate drop width. While these features are deemed important by existing literature, they may not capture all the nuanced interactions that occur, especially when drops encounter surface defects. When a drop moves over a surface with a single defect, its center velocity decreases upon encountering a defect and increases after surpassing it (Figure 1, black line). Meanwhile, the advancing and receding velocities exhibit distinct behaviors as they interact with surface defects in different ways (Figure 1, red and blue lines). Monitoring only the center velocity fails to account for these differences, limiting estimation accuracy. The advancing and receding contact lines engage with the defect differently, revealing nuanced behaviors that are not captured when considering only the center velocity.

The gap in knowledge lies in the absence of a comprehensive method that can autonomously extract and prioritize relevant features from raw video data to

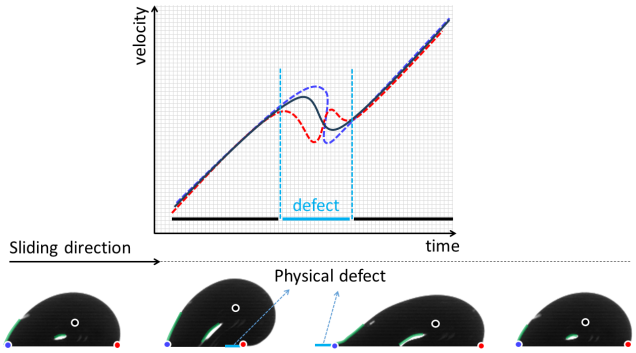

Fig. 1. Velocity profiles of a sliding drop over a surface defect. The diagram depicts the velocities at the drop's center (black), advancing edge (red), and receding edge (blue). Colors in the plot match the colored points on the schematic. As the drop interacts with the defect, the center velocity decreases, while the advancing and receding edges respond differently, revealing nuanced behaviors beyond center velocity analysis. Green curves indicate potential areas of interest for a more detailed investigation of drop dynamics.

describe the physics of sliding drops. Current models do not leverage the full potential of video data to identify subtle but important features that could enhance measurement precision, especially in challenging scenarios involving surface defects. Moreover, if we can automatically extract features, it will open up new opportunities to explore which segments of the drop contour line are crucial. For instance, we could investigate whether a combination of pixels in the drop's receding section or even the reflections within the drop itself might provide essential information on drop width (Figure 1, green drop curves). Furthermore, this method enhances estimation accuracy and increase robustness against environmental variations, including optical distortions such as minor defocusing and focus irregularities, motion blur, lighting fluctuations, as well as dust within the lenses and scattered lights that cause noise in video frames.

1.1 Main contributions

In this study, we introduced a novel deep learning approach for accurately estimating the width of sliding drops directly from side-view video data. Key outcomes and advancements of our work include:

- **Position Invariant Video Processing:** Our proposed position invariant video processing method mitigates overfitting due to positional bias while significantly reducing computational load by approximately 84%. It is applicable to scientific problems involving the motion of small objects of interest, especially when data availability is limited.
- **Low-Dimensional Absolute Position Encoding:** Our proposed ldAPE effectively addresses the anisotropic limitations commonly encountered in

conventional positional encoding methods for low-dimensional time-series data. Empirically, it outperforms both tAPE and Sin-APE on 32-dimensional data, with theoretical advantages extending up to 128 dimensions.
- **Optimized CNN-Transformer Architecture:** We developed a custom VGG8-inspired CNN architecture specifically designed for video datasets characterized by low information density. Coupled with the ConvTran time-series transformer, our model efficiently captures intricate spatiotemporal interactions. We achieved an RMSE of 48.4 µm, corresponding to a low error rate of just 1.7%. This demonstrates a considerable improvement over previous state-of-the-art models, especially in challenging scenarios involving surface defects.
- **Robustness and Interpretability:** Based on Grad-CAM visualizations, we confirmed that our model robustly identifies critical drop features, including subtle edges and reflections. This capability not only improves estimation accuracy but also enhances interpretability, offering insights into the underlying physics of drop-surface interactions.
- **Open Source Contribution:** To support future research and foster collaboration within the scientific community, we release our comprehensive sliding drop video dataset and the source code. This contribution enhances reproducibility, supports model inference, and promotes advancements in ML-based experimental fluid dynamics research.

2 Related work

2.1 Machine learning and surface science

The integration of machine learning into surface science enhances drop dynamics and contact angle analysis, improving complexity handling. Yancheshme et al. applied a random forest model to predict the behavior of impacting drops on hydrophobic and superhydrophobic surfaces [13]. Their goal was to determine the optimal conditions for inducing bouncing behavior during drop impact. They analyzed a broad set of predefined measures, including the drops' physical properties, kinematic characteristics, and surface attributes. Similarly, Zhang et al. developed a method to optimize the contact angle on rice leaf surfaces by comparing artificial neural networks (ANN) and response surface methodology (RSM) [14]. They focused on factors such as temperature, humidity, and pesticide concentration to determine the best conditions for minimizing the contact angle. ANN outperformed RSM in contact angle prediction, with pesticide concentration as the key factor. Kokalis et al. proposed a method to classify composite insulators into hydrophobicity classes using convolutional neural networks (CNNs) [15]. They used a spray method to collect images and train CNNs for insulator classification, removing human subjectivity. In the same way, Roy et al. introduced a method for detecting the hydrophobicity grade of polymeric insulators using Bi-directional Long Short-Term Memory (Bi-LSTM) classifier [16]. Rabbani et al. employed two deep learning models with fully connected dense layers to predict contact angles in tomography images of porous materials [17].

Kabir et al. used ResNet-50 to estimate contact angles, overcoming fitting limitations on hydrophilic surfaces [18]. A recent deep learning study in surface science developed a method (4S-SROF), enabling systematic analysis of sliding drops, even when occupying a small image region [19]. Shumaly et al. introduced a method based on regressions and Recurrent Neural Networks (RNNs) to estimate sliding drop width using predefined side-view features [12]. Their Long Short-Term Memory (LSTM) model demonstrated the best performance, estimating sliding drop width with a low error of 2.4% (67.6 µm RMSE), eliminating the need for cumbersome equipment while maintaining an unrestricted view of sliding drops. We now introduce more advanced end-to-end deep learning models capable of extracting features without relying on predefined physics-based measurements, enhancing accuracy to estimate sliding drop width.

2.2 Time series extrinsic regression

Time series extrinsic regression (TSER) is a regression task aimed at understanding the relationship between a time series and continuous scalar variables. Although numerous papers are published annually on time series classification [20, 21] and time series forecasting [22–24], time series extrinsic regression has received limited attention [25]. In this study, we address a TSER problem, reconstructing a time series (front-view) from a set of time series (side-view). Our approach employs a machine learning framework, formulating the task as a regression problem where the input consists of consecutive drop images and the output is a scalar value. Regression involves predicting a continuous numeric value based on a set of input features [26]. However, our goal is to estimate values that may extend the input time series or be indirect to it, without being restricted to future values.

Similar studies on regression involve estimating heart rate based on data gathered from accelerometers [27, 28]. Random Convolutional Kernel Transform (ROCKET) has demonstrated state-of-the-art results in various time series tasks by leveraging a set of random convolutional kernels to extract informative features efficiently [29]. InceptionTime, a deep learning-based approach inspired by the Inception architecture, enhances feature extraction, making it effective for capturing both short- and long-term temporal dependencies [30]. Similarly, Transformer for Time Series (TST) has been proposed as an attention-based model that excels in capturing intricate relationships within time series data by leveraging self-attention mechanisms [31]. ConvTran, a convolutional transformer model, has recently gained recognition. By combining convolutional feature extraction with transformer-based sequence modeling, ConvTran achieves superior performance in handling both local and global dependencies, making it particularly well-suited for tasks like TSER [32].

3 Materials and methods

3.1 Data collection

The sliding drop setup consists of a high-speed camera with a telecentric lens to record drop motion under uniform backlighting. Two parallel mirrors capture the front view by reflecting the backlight. The entire optical system is mounted on a rotatable breadboard to maintain alignment. Distilled water drops (32 μl) are deposited onto a tilted plane using a peristaltic pump connected to a grounded syringe needle. The technical details and a schematic of the setup and sample preparation are presented in Supplementary Information (SI) Sections S.1, and S.2. Installing the mirrors restricted the focus of the front-view camera to the last $\approx$1.5 cm of the slide path. Data was collected only within this region. Therefore, defects were fabricated on the last centimeter of the samples. The dataset was filtered to include videos with 20–250 frames for consistency. The dataset consists of 235 videos with a resolution of 1280×1024 pixels, containing a total of 11,944 frames. The number of frames per video varies depending on the drop velocity.

3.2 Data augmentation

We applied data augmentation to mimic real-world imaging variations and enhance robustness. The techniques included brightness adjustment, Gaussian blur filtering, and artifact generation. Brightness adjustment varied image intensity by $\pm$15% to account for ambient fluctuations. Gaussian blur was applied with randomly selected kernel sizes (1×1, 3×3, 5×5) to simulate defocusing and motion blur. Image artifacts were introduced as irregular stains and radiance spots to mimic lens smudges and reflections. Irregular stains were generated using sinusoidal perturbations on random circular shapes, followed by transformations such as stretching, rotation, and scaling. Radiance spots were simulated using Canny edge detection to localize drop edges, followed by circular gradient overlays. More details and pseudo-codes are provided in SI Section S.3.

3.3 Position invariant video processing methodology

Captured high-speed video frames of sliding drops have a resolution of 1280×1024 pixels. In our dataset, the largest drops reach 216×99 pixels. An initial approach involved cropping frames to 1280×99 pixels, preserving the drop's horizontal path while removing unnecessary upper and lower portions (Figure 2a). However, this approach introduced several challenges.

Firstly, the drop occupies only a small fraction of the cropped frame, leaving extensive empty space. Secondly, the model may overfit by associating drops with their absolute positions in the image rather than focusing on their shape and velocity, which are the relevant features. For instance, surface defects are always located in the last centimeter of the sliding path due to video capture constraints [12]. This carries the risk that the model becomes too closely adapted

to the droplet's dynamic behavior at a specific location, thereby limiting its ability to generalize to defects appearing at other positions.

To address these challenges, we introduced a 3D sliding window centered on the drop, which we call the sliding spatiotemporal window (SSW). We set the window size to 216×99 pixels, matching the maximum observed drop dimensions (Figure 2b). This window follows the drop's movement, keeping it centered in the frame and reducing irrelevant background. The impact of input tensor size on memory usage and computation time was obtained using a dummy input. It assesses the general computational footprint of the model's forward pass. The total memory usage M and total time T were computed as follows:

$$M = \sum_{i=1}^{n} S_i, \quad T = \sum_{i=1}^{n} t_i, \tag{3}$$

Here, S_i and t_i represent the memory usage (in bytes) and time (in milliseconds) of the i-th operation, respectively. Two experiments were conducted with different input tensor sizes: (216×99) notated as "SSW", and (1280×99) notated as "original". The percentage reduction in memory usage and computation time was computed as

$$\Delta M\% = \left(1 - \frac{M_{\text{ssw}}}{M_{\text{original}}}\right) \times 100\% = \left(1 - \frac{1796.8 \text{ MB}}{10153.8 \text{ MB}}\right) \times 100\% \approx 82.3\%, \tag{4}$$

$$\Delta T\% = \left(1 - \frac{T_{\text{ssw}}}{T_{\text{original}}}\right) \times 100\% = \left(1 - \frac{239.5 \text{ ms}}{1518.1 \text{ ms}}\right) \times 100\% \approx 84.2\%. \tag{5}$$

These results indicate that reducing the input tensor size led to an approximately 82% decrease in memory usage and an 84% decrease in computation time, while the number of model parameters remained unchanged.

Capturing temporal dynamics is essential for accurate drop width estimation. To track the drop's movement over time, we set the sequence length to 20 frames, meaning each model input consists of 20 consecutive frames with the drop centered within the SSW. Studies show that 20-frame sequences effectively capture key drop dynamics without overloading the model [12]. In general, frames 1 to 9 correspond to the past relative to the target frame (frame 10), whose width we aim to estimate, while frames 11 to 20 represent its future.

However, centering drop images inadvertently removes the drop's relative positional information within the sequence, which carries valuable temporal cues about its motion. To retain motion cues, we tracked the drop's center relative to its start. However, directly including the drop's center position could lead the model to overfit to absolute drop locations. To avoid this, we incorporated the first derivative of the drop's position with respect to time, which corresponds to its velocity. We approximated the velocity using a first-order finite difference. Specifically, we calculated it as $v_t = (x_t - x_{t-1})/\Delta t$, where x_t is the horizontal position of the drop in frame t, and Δt is the time interval between frames. The resulting velocity time series was added as an input to the model. This

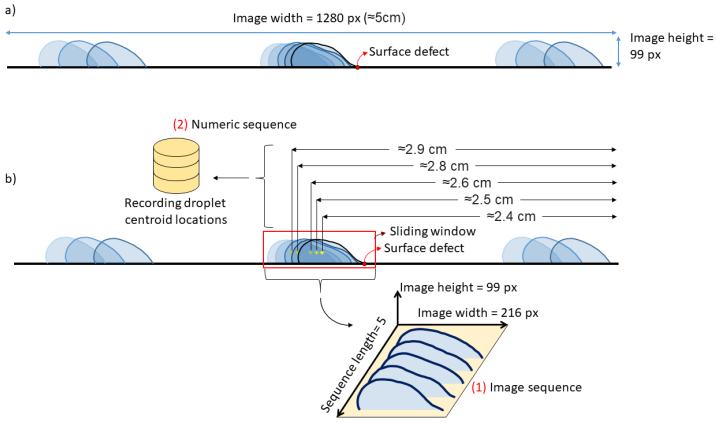

Fig. 2. Data preparation and pipeline for formatting input for the model. a) Initial approach: Cropping the full sliding path (1280×99) results in extensive empty space and positional bias due to the drop's varying location. b) Improved method: Using a SSW of size 216×99 pixels, matching the maximum drop dimensions. For demonstration, a 5-frame sequence is shown, while the model utilizes 20 frames for effective drop analysis.

helped us retain temporal motion cues while removing the risk of overfitting to absolute drop positions. Incorporating the velocity time series serves two key purposes. First, velocity is crucial for understanding drop dynamics, as it reflects frictional forces, surface interactions, and acceleration. Most importantly, with a fixed frame rate, velocity encodes positional changes and establishes a temporal link between frames.

Our approach ensures that the model focuses on the drop's shape and motion rather than its position. Additionally, it extracts only the drop region (216×99) from the original frame (1280×99), achieving an 84% reduction in computation time. We refer to this approach as position invariant video processing.

3.4 Spatiotemporal model architecture

The model begins with a VGG-style 2D CNN to extract spatial features from consecutive video frames (Figure 3). The architecture is adapted for smaller datasets and images with lower informational density. It is inspired by VGG8, but replaces standard pooling layers with BlurPooling and employs the Gaussian Error Linear Unit (GELU) as the activation function. BlurPooling improves shift-equivariance, leading to better generalization [36]. It consists of four convolutional blocks with 64, 128, 256, and 512 filters, each featuring a 3×3 convolution (padding = 1). We refer to this architecture as BlurVGG8. The extracted spatial features are reshaped to align with the temporal data. Velocity from the 4S-SROF method [19] is processed through a fully connected layer for dimensional consistency before being integrated element-wise with spatial features. The position invariant video processing method stacks consecutive drop images,

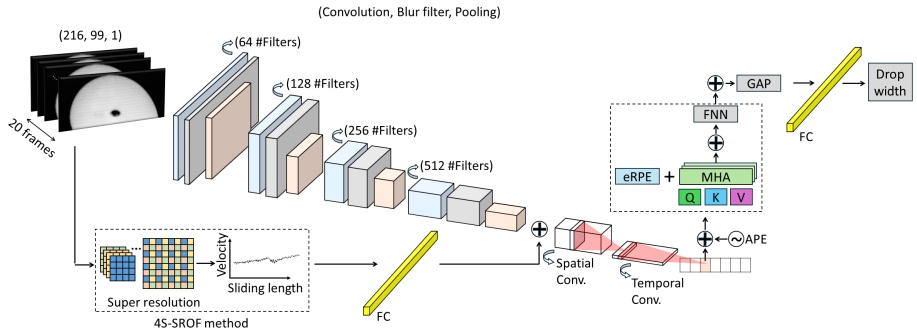

Fig. 3. Architecture of the spatiotemporal model. BlurVGG8 extracts spatial features using four convolutional blocks. Temporal dynamics are preserved by integrating velocity data with spatial features. The ConvTran architecture refines these features with additional convolutions, position encoding (APE and eRPE), and a Transformer encoder to capture long-range dependencies.

but to retain relative positional information, it requires integrating the velocity to preserve temporal dynamics.

The velocity encoded data is processed by the ConvTran architecture, starting with a spatial convolutional layer that enhances extracted features. The embedding size is set to 64, followed by temporal convolutional layers that refine short-term dependencies. Next, position encoding is applied to enhance temporal awareness. We introduce an improved Absolute Position Encoding (APE), called Low-Dimensional Absolute Position Encodings (ldAPE), to enhance the model's capability. Simultaneously, Efficient Relative Position Encoding (eRPE) captures relative frame distances. Since transformers process data in parallel, explicitly encoding temporal order is essential [38]. Next, the transformer encoder was applied with self-attention to capture long-range dependencies, analyzing frame interactions and tracking drop behavior. The number of heads was set to 4, and the feed-forward dimension was adjusted to 128 for our specific task.

We set the learning rate to 0.0001, used the AdamW optimizer with a weight decay of 1×10^{-5}, and selected a batch size of 16. We split the dataset into training, testing, and validation sets with a 60%/20%/20% distribution. The model was trained to minimize the Mean Squared Error (MSE) loss between predicted and actual widths. Performance was evaluated using the Root Mean Squared Error (RMSE) metric on the test set to maintain consistent units. To mitigate overfitting, the maximum training epochs were set to 400, with early stopping triggered by validation loss. All experiments were performed on a high-performance computing system with a single node featuring 36 CPU cores, 250 GB of memory, and an Nvidia A100 GPU.

3.5 Low-dimensional absolute position encodings

In transformer architectures, the self-attention mechanism alone cannot capture the natural order of sequential data. However, preserving the order of the sequence is crucial for accurate analysis, especially when dealing with time-series data. To overcome this limitation, transformer-based models introduce positional encoding, which injects order-related information into the input representation. The positional encoding ensures that the model can distinguish between different positions in the sequence and maintain the relationships. There are different types of positional encoding such as absolute positional encoding (APE) and relative positional encoding (RPE) as the most common techniques [39, 40].

In the APE method, absolute position information is directly incorporated into the input embedding. This is achieved by adding a position-specific encoding to each input vector, formulated as:

$$x_i = x_i + p_i \tag{6}$$

Here, $p_i \in R^{d_{\text{model}}}$ represents the positional embedding corresponding to position i, and x_i denotes the input embedding at that position. d_{model} refers to the dimension of the model's hidden representations. The positional embedding is typically defined using sine and cosine functions as follows:

$$p_i(2k) = \sin(i\omega_k), \quad p_i(2k+1) = \cos(i\omega_k) \tag{7}$$

where

$$\omega_k = 10000^{-2k/d_{\text{model}}} \tag{8}$$

While i and k are both indices, i corresponds to the feature dimension, and k is the index of the frequency components. This method (called Sin-APE) has been widely used in transformer-based architectures [38]. Sin-APE was originally proposed for language modeling, where high embedding dimensions such as 512 or 1024 are typically used. However, it exhibits limitations when applied to time series data. In low embedding dimensions, the dot product between position encodings does not consistently decrease with increasing positional distance, leading to the loss of the distance awareness property. To address this issue, time Absolute Positional Encoding (tAPE) has been introduced [32]. This method modifies the frequency term to account for both the embedding dimension d_{model} and the sequence length L, ensuring a more balanced frequency distribution:

$$\omega_k^{\text{tAPE}} = \omega_k \cdot \frac{d_{\text{model}}}{L} \tag{9}$$

Here, L is the total length of the time series.

We modified the absolute positional encoding by adjusting the frequency term to improve accuracy. The new formulation is given by:

$$\omega_k^{\text{ldAPE}} = 35^{-2k/d_{\text{model}}} \cdot \frac{2\sqrt{d_{\text{model}}}}{L} \tag{10}$$

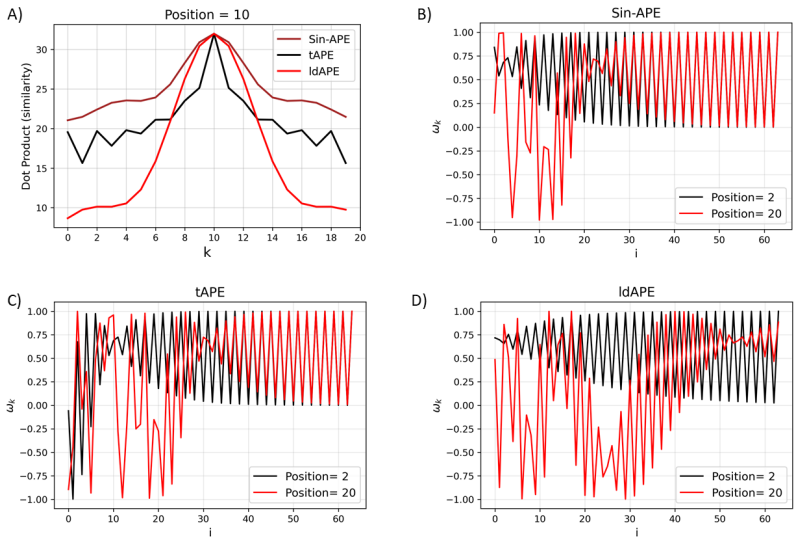

Fig. 4. Comparing different absolute positional embeddings. a) Dot product of absolute positional embeddings, demonstrating the wider similarity axis coverage in ldAPE with reduced fluctuations. b–d) Absolute positional encoding curves for positions 2 and 20 in a sequence of length 20 for Sin-APE, tAPE, and ldAPE, respectively, highlighting the improved position distinguishability in ldAPE.

This adjustment introduces a scaling factor that accounts for both d_{model}, and L. By modifying ω_k, the encoding achieves a more balanced frequency distribution, and enhancing the model's ability to distinguish between positional embeddings. We refer to this method as low-dimensional Absolute Positional Encoding (ldAPE). The dot product between positional embeddings at a fixed reference position reflects their similarity. Compared to other methods, ldAPE produces a broader and more distinct, yet smooth and noise-free, distribution of similarity scores across positions (Figure 4a), enhancing the model's ability to differentiate between them. Also, in ldAPE, the positional encodings for positions 2 and 20 show minimal overlap, indicating that ldAPE enhances position distinguishability more effectively than other methods (Figure 4b-d). The ldAPE demonstrates a better dot product than the other mentioned APEs for dimensions below 128, SI Section S.4.

4 Results and discussion

We tested LSTM models with 64, 128, and 256 units, as well as Bi-LSTM models with the same configurations. The Bi-LSTM models consistently outperformed the LSTM models, prompting us to use Bi-LSTM architectures for further experiments.

Table 1. Model comparison based on RMSE. Results are repeated over three independent runs for reliability.

Model Configuration	RMSE Avg.	RMSE std.
ViT + transformer encoder	148.2	3.6
VGG16 + BiLSTM	204.1	2.8
Pre-trained VGG16 + BiLSTM	81.9	9.1
VGG8 + BiLSTM	63.5	2.8
Pre-trained VGG8 + BiLSTM	86.1	4.0
BlurVGG8 + BiLSTM + Self attention	54.1	4.0
BlurVGG8 + ConvTran (ours)	**48.4**	2.4

Initially, we tested transformer models and the VGG16 architecture, known for their effectiveness in capturing complex patterns and features across various tasks [34, 35]. However, due to the limited amount of training data available and low information density image frames, these models did not perform as well as expected (Table 1). The concept of low information density has been used to compare information density in computer vision and Natural Language Processing (NLP), suggesting that pixels in computer vision contain less information than words in NLP [37]. Additionally, different regions of an image contribute unequally to its overall meaning. Based on this, we argue that our images have even lower information density than typical computer vision images, as only the drop contour is relevant while the rest of the image holds minimal significance. To address this, we switched to VGG8, a streamlined version of the VGG architecture with lower complexity. This change achieved an RMSE of 63.54 μm, surpassing earlier studies that used features based on domain knowledge (RMSE of 67.6 μm [12]).

Performance improved even more after modifying VGG8, replacing standard pooling with BlurPooling, utilizing Gaussian Error Linear Unit (GELU) activation, and adding self-attention after the Bi-LSTM layer, achieving an RMSE of 54.13 μm.

The modified VGG8 (BlurVGG8) was retained because it yielded better results, while ConvTran was used for the temporal component. To conduct an ablation study, three different APE variants were evaluated: Sin-APE, tAPE, and the proposed ldAPE (see Sec. 3.5). The ldAPE achieved the best performance, reaching an RMSE of 48.4μm (Table 2). To further assess the contribution of input velocity and the proposed BlurVGG8 architecture, we removed the velocity input and replaced BlurVGG8 with the original VGG8 in the best-performing configuration, observing the corresponding performance drop in each case.

Surface defects and their larger geometry create more complex time series patterns, increasing the error rate. One defect-free sample (I) and three samples with a block defect (800 μm thick) from the test set are visualized in Figure 5a: sample II (1000 $\times$ 106 μm), sample III (2000 $\times$ 74 μm), and sample IV (3000 $\times$ 174 μm). In nearly all cases, the error rate decreased compared to the previous study that used predefined features. Specifically, the error changed

Table 2. Ablation study on the effects of BlurVGG8, velocity input, and different APE variants.

Configuration	RMSE Avg.	RMSE std.
BlurVGG8 + ConvTran (Sin-APE)	53.8	6.1
BlurVGG8 + ConvTran (tAPE)	50.2	4.3
BlurVGG8 + ConvTran (ldAPE)	**48.4**	2.4
BlurVGG8 + ConvTran (ldAPE) – without velocity	61.1	4.9
VGG8 + ConvTran (ldAPE)	57.5	4.1

from 30.8 μm to 33.9 μm for sample I, from 56.2 μm to 21.9 μm for sample II, from 50.4 μm to 49.3 μm for sample III, and from 82.8 μm to 57.1 μm for sample IV [12].

Additionally, to evaluate the generalization capability, we compared their results on a sliding drop example that was not part of the training dataset. This experiment was performed on a hydrophobic surface (PFOTS_Si) with a block defect (800 μm thick, 3000 μm long, 23 μm high). While PFOTS_Si surfaces were in the training videos, this specific defect size was not. During the experiment, the drop stuck to the defect and detached very slowly, which had not occurred in the dataset. The model with predefined features based on domain knowledge produced an RMSE of 112.5 μm [12], while the model utilizing auto-extracted features achieved a significantly lower RMSE of 66.6 μm (Figure 5b). We hypothesized that deep learning models with automated feature extraction would better capture complexities than those using predefined features. The RMSE improvement confirmed this. We altered the frames by adjusting illumination and introducing blurriness and artifacts, simulating challenging real-world conditions. The results indicated that the model's estimations remained robust under these perturbations, exhibiting minor fluctuations and slight deviations in the drop width measurements (Figure 5c).

Feature sensitivity. To evaluate how the model identifies key features for drop width estimation, we applied the Grad-CAM algorithm to visualize the Regions of Interest (ROIs) in the input images (Figure 6). The figure presents seven middle frames from a sequence of 20, focusing on estimating the width of the central frame (frame 10). Each row represents a different time step in the video sequence, illustrating how the model's attention dynamically shifts as the sliding drop interacts with the surface.

The sequence captures the critical moment when the advancing edge of the drop encounters a surface defect (Figure 6, frame 7) and its subsequent response. The heatmaps in the middle column are spatially normalized between 0 and 1, ensuring that the most significant regions within each frame are distinctly highlighted. These visualizations reveal that the model consistently focuses on the drop's contour. The heatmaps in the left column remain unnormalized, preserving absolute activation values to capture spatiotemporal dependencies across frames. This enables a direct comparison of activation patterns over time. Notably, frames 8 and 9 exhibit the strongest activations, suggesting they provide

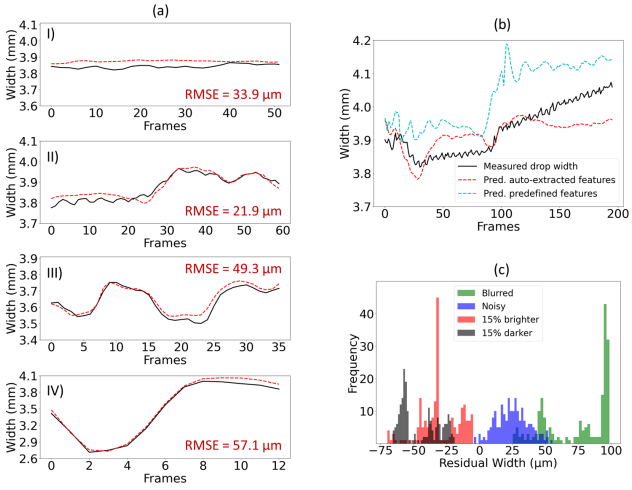

Fig. 5. a) Drop width measurements while sliding over a defect-free surface and three samples with defect, all 800 μm thick: sample II (length 1000 μm, height 106 μm), sample III (2000 $\times$ 74 μm), and sample IV (3000 $\times$ 174 μm). b) Comparison of the predefined features model and the automated feature extraction model on a sample outside the training dataset, with RMSEs of 112.5 μm and 66.6 μm, respectively. c) Effect of data augmentations (illumination changes, blurriness, and artifacts) on estimation diagrams using the automated feature extraction model. Distribution of residual errors (predicted - measured width) under different data augmentations. Each individual bar corresponds to the frequency of a specific residual value range.

the most critical information for accurately estimating the width of frame 10. This experiment demonstrates that the model effectively identifies key features aligned with established domain knowledge, such as drop length, height, and receding. Additionally, the Grad-CAM visualizations highlight the model's dynamic attention shifts, particularly at critical interaction points, reinforcing its ability to capture spatiotemporal dependencies. This opens the door for further studies to explore deeper feature correlations and refine automated methods for analyzing sliding drops.

5 Conclusions

In this study, we introduced a novel position invariant video processing method that effectively prevents overfitting to object location while reducing computation time by 84%. This is achieved by introducing the sliding spatiotemporal window (SSW) concept and incorporating the first derivative of the position of the region of interest into an architecture capable of processing both spatial and temporal data. The approach is scalable and can be extended to higher-dimensional cases, such as 2D object motion. Our approach, which leverages both a specialized VGG8-inspired architecture and the novel ldAPE representation, is well-suited for addressing spatiotemporal challenges with low information density, such as drop motion analysis. This method can

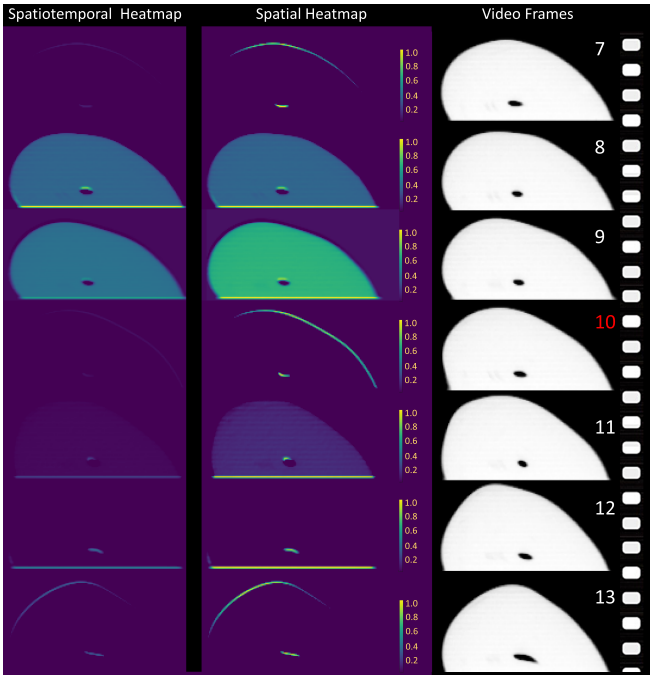

Fig. 6. Grad-CAM visualization of key regions influencing drop width estimation. The normalized heatmaps (middle column) emphasize critical spatial features, primarily the drop's edges, while the unnormalized heatmaps (left column) preserve absolute activation values, capturing spatiotemporal dependencies across frames.

effectively address challenges in drop and soft matter research. It is also applicable to scientific domains involving video sequences where the temporal contour evolution of small objects of interest is critical and data availability is limited, such as in biomedical video analysis. Moreover, it integrates with interpretability techniques like Grad-CAM, offering deeper insights into model behavior by highlighting the most influential video features. The interpretability and performance of our method pave the way for uncovering new correlations. For example, we observed that subtle reflections within drops, although seemingly insignificant, may carry meaningful information about drop geometry. Our dataset includes variations in drop viscosity, surface chemistry, wettability, sliding angle, and surface defect geometry, enabling our research to address a broad range of physical conditions and support generalization. However, the current scope does not include phenomena such as slide electrification or extreme wetting regimes (e.g., superhydrophobic or superhydrophilic surfaces), which are left for future investigation. This approach is currently being applied in experimental workflows at the Max Planck Institute for Polymer Research to support automated drop analysis in surface science experiments. To support further research, we have made our code and dataset publicly available.

6 Acknowledgments.

We thank Geoff Webb for the valuable scientific discussions. We acknowledge financial support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 883631) (S.S., F.D., H.-J.B.). Additional support was provided by the Priority Programme 2171 Dynamic Wetting of Flexible, Adaptive, and Switchable Surfaces (Grant Nos. BU 1556/36 and BE 3286/6-1: H.-J.B., R.B.), and by the German Research Foundation (DFG) through the Collaborative Research Center (CRC) 1194 – Interaction between Transport and Wetting Processes (Project-ID 265191195), project C07N and T02 (R.B., H.-J.B.).

7 Conflict of Interest

The authors declare no competing interests.

8 Supplementary Information

Several related works and additional implementation details are discussed in the Supplementary Information document, where the following references are also cited [12, 33, 44].

References

1. Sbragaglia M, Biferale L, Amati G, Varagnolo S, Ferraro D, Mistura G, Pierno M. Sliding drops across alternating hydrophobic and hydrophilic stripes. Physical review E. 2014 Jan;89(1):012406.
2. Yonemoto Y, Suzuki S, Uenomachi S, Kunugi T. Sliding behaviour of water-ethanol mixture droplets on inclined low-surface-energy solid. International Journal of Heat and Mass Transfer. 2018 May 1;120:1315-24.
3. Li X, Bodziony F, Yin M, Marschall H, Berger R, Butt HJ. Kinetic drop friction. Nature communications. 2023 Jul 29;14(1):4571.
4. Extrand CW, Kumagai Y. Liquid drops on an inclined plane: the relation between contact angles, drop shape, and retentive force. Journal of colloid and interface science. 1995 Mar 15;170(2):515-21.
5. Furmidge CG. Studies at phase interfaces. I. The sliding of liquid drops on solid surfaces and a theory for spray retention. Journal of colloid science. 1962 Apr 1;17(4):309-24.
6. Larkin BK. Numerical solution of the equation of capillarity. Journal of Colloid and Interface Science. 1967 Mar 1;23(3):305-12.
7. Frenkel YI. On the behavior of liquid drops on a solid surface. 1. The sliding of drops on an inclined surface. arXiv preprint physics/0503051. 2005 Mar 7.
8. Buzágh A, Wolfram E. Bestimmung der Haftfähigkeit von Flüssigkeiten an festen Körpern mit der Abreißwinkelmethode. II. Kolloid-Zeitschrift. 1958 Mar;157:50-3.
9. Extrand CW, Gent AN. Retention of liquid drops by solid surfaces. Journal of colloid and interface science. 1990 Sep 1;138(2):431-42.

10. Gao N, Geyer F, Pilat DW, Wooh S, Vollmer D, Butt HJ, Berger R. How drops start sliding over solid surfaces. Nature Physics. 2018 Feb 1;14(2):191-6.
11. Li X, Bista P, Stetten AZ, Bonart H, Schür MT, Hardt S, Bodziony F, Marschall H, Saal A, Deng X, Berger R. Spontaneous charging affects the motion of sliding drops. Nature Physics. 2022 Jun;18(6):713-9.
12. Shumaly S, Darvish F, Li X, Kukharenko O, Steffen W, Guo Y, Butt HJ, Berger R. Estimating sliding drop width via side-view features using recurrent neural networks. Scientific Reports. 2024 May 27;14(1):12033.
13. Yancheshme AA, Hassantabar S, Maghsoudi K, Keshavarzi S, Jafari R, Momen G. Integration of experimental analysis and machine learning to predict drop behavior on superhydrophobic surfaces. Chemical Engineering Journal. 2021 Aug 1;417:127898.
14. Zhang J, Lin G, Yin X, Zeng J, Wen S, Lan Y. Application of artificial neural network (ANN) and response surface methodology (RSM) for modeling and optimization of the contact angle of rice leaf surfaces. Acta physiologiae plantarum. 2020 Apr;42:1-5.
15. Kokalis CC, Tasakos T, Kontargyri VT, Siolas G, Gonos IF. Hydrophobicity classification of composite insulators based on convolutional neural networks. Engineering Applications of Artificial Intelligence. 2020 May 1;91:103613.
16. Roy SS, Paramane A, Singh J, Chatterjee S. Accurate Hydrophobicity Grade Detection of Polymeric Insulators in Extremely Wetted and Humid Environments Using Bi-LSTM Neural Network Classifier. In2022 IEEE Power & Energy Society General Meeting (PESGM) 2022 Jul 17 (pp. 1-5). IEEE.
17. Rabbani A, Sun C, Babaei M, Niasar VJ, Armstrong RT, Mostaghimi P. DeepAngle: Fast calculation of contact angles in tomography images using deep learning. Geoenergy Science and Engineering. 2023 Aug 1;227:211807.
18. Kabir H, Garg N. Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements. Scientific reports. 2023 Jan 27;13(1):1497.
19. Shumaly S, Darvish F, Li X, Saal A, Hinduja C, Steffen W, Kukharenko O, Butt HJ, Berger R. Deep learning to analyze sliding drops. Langmuir. 2023 Jan 12;39(3):1111-22.
20. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. Data mining and knowledge discovery. 2019 Jul;33(4):917-63.
21. Faouzi J. Time series classification: A review of algorithms and implementations. Machine Learning (Emerging Trends and Applications). 2022.
22. Lim B, Zohren S. Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A. 2021 Apr 5;379(2194):20200209.
23. Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A. Deep learning for time series forecasting: a survey. Big data. 2021 Feb 1;9(1):3-21.
24. Benidis K, Rangapuram SS, Flunkert V, Wang Y, Maddix D, Turkmen C, Gasthaus J, Bohlke-Schneider M, Salinas D, Stella L, Aubet FX. Deep learning for time series forecasting: Tutorial and literature survey. ACM Computing Surveys. 2022 Dec 7;55(6):1-36.
25. Tan CW, Bergmeir C, Petitjean F, Webb GI. Time series extrinsic regression: Predicting numeric values from time series data. Data Mining and Knowledge Discovery. 2021 May;35(3):1032-60.
26. Sammut C, Webb GI, editors. Encyclopedia of machine learning. Springer Science & Business Media; 2011 Mar 28.

27. Reiss A, Indlekofer I, Schmidt P, Van Laerhoven K. Deep PPG: Large-scale heart rate estimation with convolutional neural networks. Sensors. 2019 Jul 12;19(14):3079.
28. Zhang Z, Pi Z, Liu B. TROIKA: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. IEEE Transactions on biomedical engineering. 2014 Sep 19;62(2):522-31.
29. Dempster A, Petitjean F, Webb GI. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery. 2020 Sep;34(5):1454-95.
30. Ismail Fawaz H, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller PA, Petitjean F. Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery. 2020 Nov;34(6):1936-62.
31. Mohammadi Farsani R, Pazouki E. A transformer self-attention model for time series forecasting. Journal of Electrical and Computer Engineering Innovations (JECEI). 2020 Nov 21;9(1):1-0.
32. Foumani NM, Tan CW, Webb GI, Salehi M. Improving position encoding of transformers for multivariate time series classification. Data mining and knowledge discovery. 2024 Jan;38(1):22-48.
33. Canny J. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. 1986 Nov(6):679-98.
34. Wen Q, Zhou T, Zhang C, Chen W, Ma Z, Yan J, Sun L. Transformers in time series: A survey. arXiv preprint arXiv:2202.07125. 2022 Feb 15.
35. Jiang ZP, Liu YY, Shao ZE, Huang KW. An improved VGG16 model for pneumonia image classification. Applied Sciences. 2021 Nov 25;11(23):11185.
36. Zhang R. Making convolutional networks shift-invariant again. InInternational conference on machine learning 2019 May 24 (pp. 7324-7334). PMLR.
37. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R. Masked autoencoders are scalable vision learners. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2022 (pp. 16000-16009).
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in neural information processing systems. 2017;30.
39. Wu K, Peng H, Chen M, Fu J, Chao H. Rethinking and improving relative position encoding for vision transformer. InProceedings of the IEEE/CVF international conference on computer vision 2021 (pp. 10033-10041).
40. Dufter P, Schmitt M, Schütze H. Position information in transformers: An overview. Computational Linguistics. 2022 Sep 1;48(3):733-63.
41. Boinovich LB, Emelyanenko AM. Recent progress in understanding the anti-icing behavior of materials. Advances in Colloid and Interface Science. 2024 Jan 1;323:103057.
42. Ghasemlou M, Stewart C, Jafarzadeh S, Dokouhaki M, Mathesh M, Naebe M, Barrow CJ. Self-lubricated, liquid-like omniphobic polymer brushes: Advances and strategies for enhanced fluid and solid control. Progress in Polymer Science. 2025 Feb 19:101933.
43. Zhou X, Wang Y, Li X, Sudersan P, Amann-Winkel K, Koynov K, Nagata Y, Berger R, Butt HJ. Thickness of Nanoscale Poly (Dimethylsiloxane) Layers Determines the Motion of Sliding Water Drops. Advanced Materials. 2024 Jul;36(29):2311470.
44. Darvish F, Shumaly S, Li X, Dong Y, Diaz D, Khani M, Vollmer D, Butt HJ. Control of spontaneous charging of sliding water drops by plasma-surface treatment. Scientific Reports. 2024 May 9;14(1):10640.