

# Exploring Deep Learning and Data Representations for the Prediction of Erosion Channels

Alexander Vu\*

Nishal Sukumar†

Avu@clarku.edu

nnishalsukumar@clarku.edu

Clark University

Worcester, Massachusetts, USA

Arshad Kudrolli‡

Li Han§

AKudrolli@clarku.edu

LHan@clarku.edu

Clark University

Worcester, Massachusetts, USA

## ABSTRACT

Accurately forecasting the evolution of erosion channels is essential for safeguarding buildings, dams, and other earth structures. In this paper, we consider a simplified channel system and two machine learning challenges based on thousands of high-fidelity simulations: single-frame prediction of a channel's ultimate direction (left, right, or split) and long-horizon synthesis of its full trajectory from an early image or image sequence. For final direction prediction, a light-weight convolutional network and a hybrid Autoencoder-CatBoost pipeline are used, with the latter achieving better early-warning accuracy, the former remaining unexpectedly robust when inputs are aggressively down-sampled or binarized. For path prediction, a recurrent video model (PredRNN-V2) excels at intersection-over-union, whereas a latent-space autoencoder predicts finer path geometry deeper into the future, especially for binary images. These findings quantify how model architecture, training-set size, and image fidelity affect actionable lead time in erosion monitoring, and point to several promising research directions.

## CCS CONCEPTS

- Applied Computing → Physical Sciences and Engineering;
- Applied computing → Physics; • Computing Methodologies → Neural Networks; Supervised Learning,

## KEYWORDS

Deep Learning, Erosion Channel Direction Prediction, Erosion Path Prediction

### ACM Reference Format:

Alexander Vu, Nishal Sukumar, Arshad Kudrolli, and Li Han. 2025. Exploring Deep Learning and Data Representations for the Prediction of Erosion Channels. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Undergraduate and Masters Consortium (KDD*

\*Primary undergraduate co-author, Clark University

†Graduate co-author, Clark University

‡Faculty advisor from Department of Physics, Clark University

§Faculty advisor from Department of Computer Science, Clark University

UMC 2025) (KDD '25). ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

## 1 INTRODUCTION

Channel networks are complex systems that shape both surface and subsurface landscapes through processes of erosion, sediment transport, and deposition [1–3]. These systems, ranging from the branching patterns of river basins across the Earth's surface to intricate pathways within natural aquifers and engineered porous media, exemplify the dynamic interplay between fluid flow and geomorphological evolution [4]. While surface channels are often shaped gradually over geological timescales by persistent river currents, extreme events such as dam failures or floods can produce rapid and dramatic morphological changes. In the subsurface, fluid flow through porous media can erode and reorganize internal structures, altering hydraulic properties and connectivity. Chemical dissolution further contributes to the development of voids and channels, particularly in carbonate-rich environments, complementing mechanical erosion in shaping underground networks. The emergence and evolution of such channels are governed by a highly nonlinear interplay of hydrodynamic forces, material heterogeneity, and chemical interactions, rendering their development inherently difficult to predict.

The formation and evolution of these channel systems are influenced not only by local material evolution but also by the spatial distribution of water sources. These factors can determine the direction and intensity of flow more strongly than the local nature of the material. Despite the apparent diversity of these systems, they are governed by shared physical principles, making them amenable to unifying theoretical frameworks. A Laplacian framework, for example, has been employed to model two-dimensional groundwater flow under the Dupuit approximation [5]. This formulation has enabled the development of physics-based models capable of reproducing erosion patterns observed in controlled laboratory experiments under defined boundary conditions [6–8]. Such models offer valuable insights into the coupling between hydrodynamics and geomorphology.

Nevertheless, significant challenges remain in extending these models to natural settings, where complete boundary and initial conditions are rarely known. Developing robust, predictive models of erosion and channel formation under conditions of partial information remains a critical area of research. Addressing these challenges will advance our ability to forecast landscape evolution and optimize the design of engineered systems for water management, environmental remediation, and resource extraction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

KDD '25, August 3–7, 2025, Toronto, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0400-5/25/08

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

To study erosion at relatively low cost, physics-based simulations are widely used. Fluid motion in porous media obeys Darcy's law

$$\nabla \cdot (\kappa \nabla P) = 0, \quad (1)$$

where  $P$  is pore pressure and  $\kappa$  the evolving permeability; the local fluid velocity is  $v_f = -\kappa \nabla P$ . Erosion begins where  $|v_f|$  exceeds a critical threshold. We adopt the hybrid erosion model of Kudrolli, *et al.* [7], which partitions the porous matrix into immobile grains, mobile grains, and fluid [9]. Hydrodynamic stress mobilizes grains, which advect with the flow at reduced speed and redeposit when slowed or obstructed. Interfaces of contrasting porosity therefore concentrate erosion and seed channel growth.

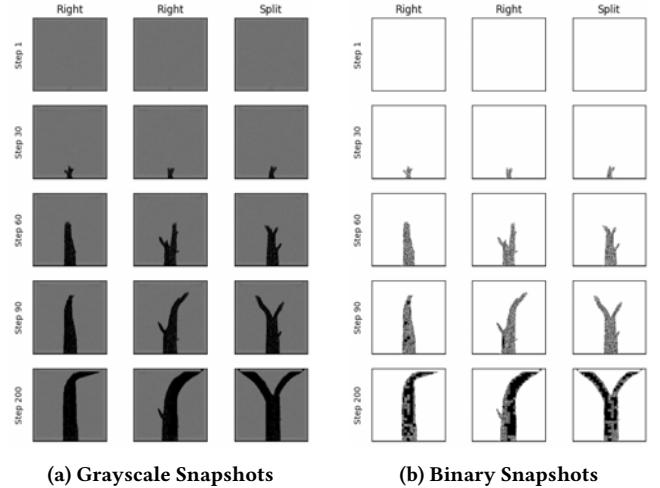
Although physics-based simulators can now generate large labeled image sets at modest cost, systematic data-driven forecasting of erosion channels is still rare, despite growing interest in applying and adapting machine learning (ML) techniques to problems involving erosion and porous media. Several ML methods were applied to identify the primary drivers of soil erosion [10]. Convolutional neural networks trained on high-resolution Unmanned Aerial Vehicle (UAV) imagery were used to classify erosion severity and analyze contributing factors such as land use, vegetation cover, and slope [11]. A comprehensive review of recent advances and ongoing challenges in ML for porous media organizes the work in six application domains: heat exchangers and thermal storage, energy storage and combustion, electrochemical systems, hydrocarbon reservoirs, carbon capture and sequestration, and groundwater [12].

While prior studies cover a broad and diverse set of problems, they differ substantially from the erosion forecasting challenges we address here. To the best of our knowledge, the most closely related prior work is the master's thesis by Lyu [13], which used a physics-based simulation model to address (i) a binary erosion type classification task (uniform vs branched patterns) and (ii) long-horizon prediction of uniform river network images with a spatiotemporal Long Short-Term Memory (SPLSTM) model. The former poses a simpler decision boundary compared with the three-way direction prediction task introduced in the next section, while the latter's impressive accuracy appears sensitive to how training and test frames were handled, hinting at possible data leakage. However, their study showed the efficacy of the specific SPLSTM model they used for the prediction of erosion networks.

To our knowledge, no published work has rigorously tested whether modern deep networks can recover both the ultimate direction and the detailed path of channel growth from early-time snapshots under a strictly leak-free protocol. We close this gap by benchmarking convolutional, hybrid autoencoder-CatBoost[14; 15], and sequence-model (PredRNN)[16] architectures on thousands of simulated channel growth images and analyzing prediction quality across data regimes.

## 2 PROBLEM FORMULATION

We study a simplified channel system in which a single conduit originates at the bottom–centre of a square porous domain and grows upward, eventually veering left, right, or splitting into both directions. The domain has two fluid inlets (top corners) and one outlet (bottom centre). Channel evolution is recorded as a sequence of 200



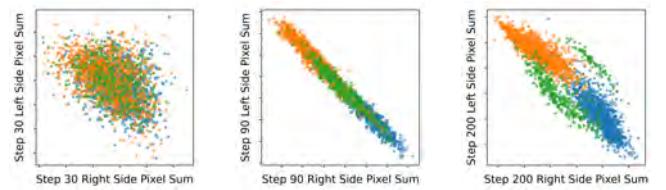
**Figure 1: Channel growth in three illustrative simulations (columns) at steps 1, 30, 60, 90, and 200 (top-bottom).**

grayscale frames, each of size  $200 \times 200$  pixels. Visible growth typically begins near frame 20 and is essentially complete by frame 110; the final frame serves as the class label for a run.

Building on this simulation framework, we consider two complementary ways to represent the image data. The first retains the raw grayscale output, preserving information about the surrounding porous matrix. The second converts each frame to a binary map that isolates channel pixels and sets the background to zero. Figure 1 illustrates three representative simulations in both formats. Which representation ultimately proves more informative is an empirical question addressed in our study.

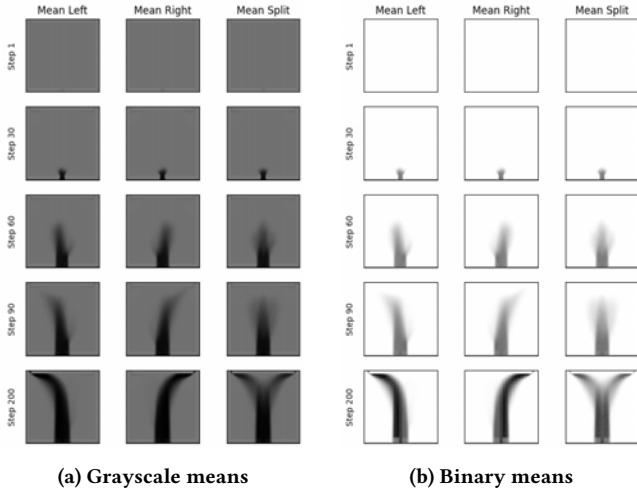
Regardless of representation, we cast channel evolution into two concrete prediction problems. **(i) Final-direction Prediction:** given a single intermediate frame, predict whether the run will end **LEFT**, **RIGHT**, or **SPLIT**. **(ii) Path forecasting:** given the first  $k$  frames, generate the channel image at any future step  $t > k$ , predicting its trajectory.

Before delving into model details, an exploratory analysis gauges how much signal these raw images contain. Figure 2 plots the total grayscale intensity in the right half of each frame against that in the left half at steps 30, 90, and 200. Early in the run the three outcome classes overlap almost completely; only by the final frame does a partial separation emerge, implying that coarse pixel-sum statistics confer little discriminative power.



**Figure 2: Left-vs-right pixel-sum scatter at steps 30, 90, and 200, by final direction: blue – left, orange – right, green-split.**

Averaging across many runs offers a second, complementary view of the signal available to a learning algorithm. Figure 3 shows mean images for the left, right, and split groups at steps 1, 30, 60, 90, and 200. Direction-specific structure becomes discernible only after step 60, underscoring the need for models that can detect subtle, early-stage cues. Collectively, these observations argue for high-capacity spatio-temporal models; Section 3 therefore details the dataset splits and the learning architectures we employ.



**Figure 3: Mean images (columns = outcome) at steps 1, 30, 60, 90, 200 (top-bottom).**

### 3 METHODS

#### 3.1 Dataset and Preprocessing

We simulate 3 000 baseline runs, each producing a sequence of 200 grayscale frames  $200 \times 200$  pixels that record the evolution of channel growth. For computational efficiency, each frame is down-sampled to  $50 \times 50$  pixels. In addition to these grayscale images, we also store binarized versions to assess how input representation and resolution affect model performance.

Each simulation is labeled by its final direction, Left (L), Right (R), or Split (S), with class frequencies of approximately 40 % L, 44 % R, and 16 % S. We divide the simulations run-wise into 1 920 for training (64 %), 480 for validation (16 %), and 600 for testing (20 %). To address the L/R class imbalance during training, each training image is augmented with its left-right mirror: this transformation swaps L and R labels while leaving S unchanged, effectively balancing the two dominant classes without inflating the minority class. Validation and test sets are left unaugmented to preserve the natural class distribution and avoid symmetry-based information leakage. We acknowledge, however, that this asymmetric treatment of mirrors may introduce a distribution shift between training and validation. A systematic study of alternative splits, such as mirrored validation folds, cross-validation at the run level, or balanced resampling of Split cases, is left to future work.

#### 3.2 Final-Direction Prediction

**Task and Evaluation Metric.** Final-direction prediction is a problem of three classes with an unbalanced label distribution (the SPLIT class is only ~16%). For potential evaluation metrics, we examined the three common aggregates of per-class F1—macro, weighted, and micro. Empirically, macro-F1 over-emphasised the rare class, while weighted-F1 was dominated by the two majority classes. We therefore adopt micro-F1, which is numerically identical to overall accuracy, as our single objective for training models and comparing their performances.

**CNN.** Because our data consists of channel images and CNN is essentially a default starting architecture for image-related machine learning problems, we begin with a compact VGG-style convolutional neural network. Specifically, the model consists of 3 VGG-blocks with 2 layers each, followed by a global average pool and a 2-layer classifier Multilayer perceptron (MLP). The loss function is a simple cross-entropy loss over the three classes. We also experimented with a full VGG architecture, both from scratch and with ImageNet pre-training, but it offered no performance gain and is therefore not included there.

**Autoencoder-CatBoost (ACB).** The idea here is to use an autoencoder [14] to generate features of the images and pass the encoded features to a CatBoost [15] classifier. The autoencoder efficiently extracts dense representations, while gradient-boosted decision trees often excel at classification when sample sizes are limited, making the hybrid approach well suited to our data.

Spatio-Temporal models such as a Convolutional LSTM [17] were also attempted for this problem. This approach makes sense since we are working with sequential image data. However, models based on this architecture performed worse than their simpler CNN counterparts and took much longer to run. Thus, their results are not discussed in this paper.

#### 3.3 Path Forecasting

**Task and Evaluation Metric.** We cast frame-to-frame forecasting as a pixel-wise regression problem and train an encoder-decoder sequence model based on PredRNN-V2 [16]. Because the task is to generate the full image at each time step, we use the mean-squared error (MSE) between the predicted frame  $\hat{I}_t$  and the ground-truth frame  $I_t$  as the loss function for our model training,

$$\mathcal{L}_{\text{MSE}} = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W (\hat{I}_t(i, j) - I_t(i, j))^2,$$

where  $H$  and  $W$  are the image height and width. MSE directly penalises per-pixel intensity deviations and is standard for video prediction.

Pixel-wise error, however, does not necessarily reflect the geometric similarity between the actual and forecasted channel paths. We therefore also consider the Intersection-over-Union (IoU) between the predicted and ground-truth channels for performance evaluation,

$$\text{IoU}(t) = \frac{|\hat{C}_t \cap C_t|}{|\hat{C}_t \cup C_t|},$$

IoU rewards shape overlap and is widely used in segmentation benchmarks, providing a complementary, geometry-aware assessment of forecasting quality.

**Autoencoder.** Based on the same encoded features as used in the direction prediction, we use a small 3-layer MLP in order to predict the next encoding rather than predict the next image, using a simple MSE loss. This simplifies calculations and reduces some of the instability we saw in other architectures.

**PredRNNv2.** The PredRNN is a Convolutional LSTM network that implements a secondary memory that is passed from layer to layer along with the image logits. This gives the model a constant understanding of the past. We implement this model with 8 filters in each of 4 layers. The model uses MSE loss combined with cosine similarity. PredRNN was chosen due to its high performance on the moving MNIST dataset.

## 4 RESULTS

### 4.1 Final Direction Prediction

Predicting the ultimate direction in which a growing channel will eventually be left, right, or split, is a key step toward any erosion channel prediction and intervention. We therefore begin our analysis with this task, training two architectures, a compact convolutional network (CNN) and an Autoencoder followed by a CatBoost (ACB), on the full set of gray-scale images at their  $50 \times 50$  pixel resolution. Because the three classes are highly imbalanced (SPLIT represents only  $\approx 16\%$ ), we follow earlier work and report micro-F1 (equivalent to overall accuracy) as the key metric.<sup>1</sup>

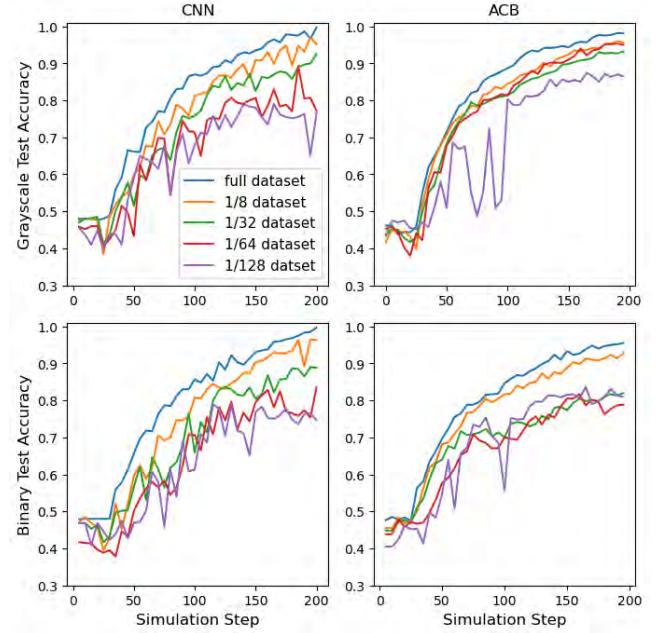
**Progress over simulation time.** Table 1 tracks micro-F1 at eight representative simulation steps. Both models start a little above random (50%) at step 30, climb to the mid-70s by step 60, and plateau above 0.85 once the channel morphology is fully developed. Throughout, ACB maintains a consistent 4–5 percentage point edge, hinting that its latent representation may capture class-specific cues more effectively than the purely convolutional pipeline. This global view sets the stage for two follow-up questions: How sensitive are these gains to the amount of training data? And which classes benefit most?

**Table 1: Micro-F1 (overall accuracy) for final-direction prediction at selected time steps.**

Step	30	35	40	60	80	100	120	200
CNN	0.49	0.56	0.59	0.73	0.81	0.87	0.89	0.99
ACB	0.55	0.62	0.64	0.78	0.85	0.89	0.94	0.98

**Impact of training-set size.** Figure 4 shows how accuracy varies as the training set is reduced from 3,840 cases to as few as  $\frac{1}{128}$  of that. For both grayscale (top) and binary (bottom) inputs, performance remains stable down to  $\frac{1}{8}$ , begins to degrade at  $\frac{1}{32}$ , and drops sharply at  $\frac{1}{64}$  and below. Across most settings, ACB maintains a 4–5 point lead over CNN, particularly in the early-warning window (steps 40–80). At the extreme low-data regime ( $\frac{1}{128}$ ), however, ACB deteriorates more abruptly—its accuracy stays near chance until late in the simulation, when performance improves slightly due to memorized pattern matching.

<sup>1</sup>A systematic comparison against macro-F1, weighted-F1, and precision-recall AUC is left for future work; such a study would clarify how robust each model is to changes in cost assumptions.



**Figure 4: Micro-F1 versus simulation step as the amount of training data is reduced.**

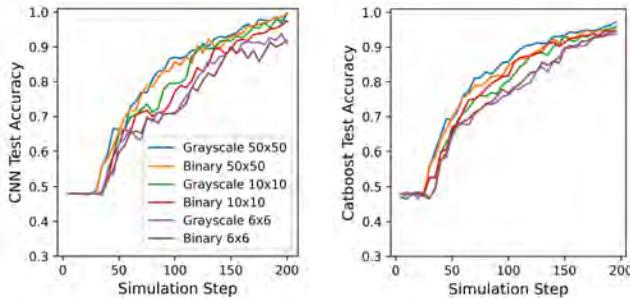
**Class-by-class performance.** To understand which directions are hardest to predict, Table 2 presents confusion matrices for both models at step 35, a time when performance is past the purely random regime yet early enough to be operationally useful. The two majority classes, LEFT and RIGHT, account for over 90% of the test set and are recognized with reasonable accuracy. The minority SPLIT class, by contrast, remains essentially invisible, a direct consequence of the skewed prior and a signal that additional balancing strategies may be required. The SPLIT class gradually becomes distinguishable in later steps, with accuracy values improving and ACB generally outperforming CNN.

**Table 2: Confusion matrices at step 35 (50 × 50 gray-scale images).**

Pred.	CNN: Actual			ACB: Actual		
	Left	Right	Split	Left	Right	Split
Left	129	81	47	158	74	47
Right	85	207	51	55	213	51
Split	0	0	0	1	1	0
Per-class ACC.	0.60	0.72	0.00	0.62	0.74	0.00

**Effect of image resolution.** Another practical lever is the spatial resolution of each frame. Figure 5 plots accuracy over training steps for the input from  $50 \times 50$ , down to  $10 \times 10$  and an extreme  $6 \times 6$ , in both grey-scale and binary form.

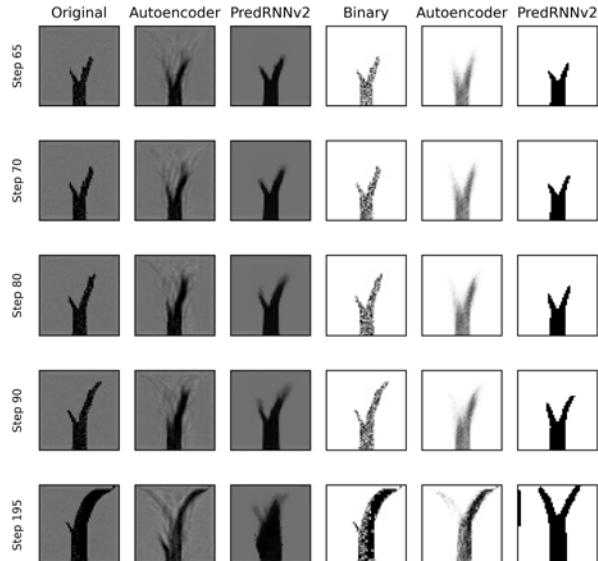
At early steps ( $\leq 60$ ) resolution matters: the  $6 \times 6$  curves sit 10–12 pp below the baseline for the CNN and about 6 pp for ACB. A  $10 \times 10$  grid, however, already recovers some of the loss, trailing



**Figure 5: Micro-F<sub>1</sub> versus simulation step for different spatial resolutions. Left – CNN; right – ACB.**

by only 3–4 pp. Late in the simulation, the gap all but closes, except for the  $6 \times 6$  CNN, which finishes roughly 5 pp below the accuracy of full resolution. Overall, ACB has shown more robustness with respect to spatial resolution, and having higher resolutions helps with early-stage predictions.

## 4.2 Path Forecasting

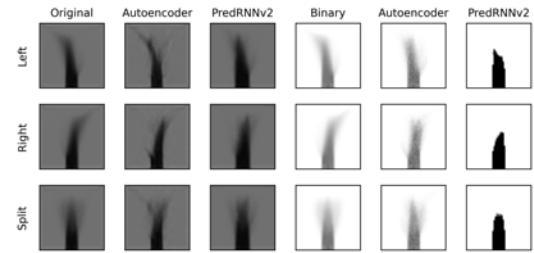


**Figure 6: Example Channel Path Predictions, with step 60 being the last information provided to the models**

Figure 6 offers a representative, but favorable, illustration of our four forecasting models. After receiving frames up to step 60, every model reproduces the right-branching trajectory through step 90, yet each exhibits its own traits: the autoencoder forecasts (columns 2 & 5 for grayscale and binary images) preserve the forked geometry but blur fine detail, whereas the PredRNN-V2 forecasts (columns 3 & 6) show a thickened central trunk with sharp tips. A significant failure appears in the binary PredRNN-V2 forecast at step 195, which drifts into a full Y-split instead of the single

rightward branch maintained by the other three models. The cause for this requires further investigation.

To visually assess overall path prediction performance, we aggregated the forecasts from all test cases, averaging those that ultimately go left, right, or split, and presented these per-class means in Figs. 7 and 8. At step 90, both grayscale models preserve the dominant branch direction: their means for ‘left’ and ‘right’ tilt correctly. The mean of the actual split images (bottom left corner) shows a faint Y-shaped fork. The autoencoder preserves this bifurcation, albeit with slightly thinner arms, whereas PredRNN-V2 collapses the pattern into a single, broad central trunk. There are also differences in shape details: the grayscale autoencoder (second column) keeps a tapered trunk and hints of secondary fingers, whereas the grayscale PredRNN-V2 (third column) produces a noticeably thicker, blurrier stem that begins to erode lateral detail. The same tendencies appear, but are amplified, in the binary models (right half of each panel): the autoencoder’s branches fade, and the binary PredRNN collapses much of the fork into a solid block.

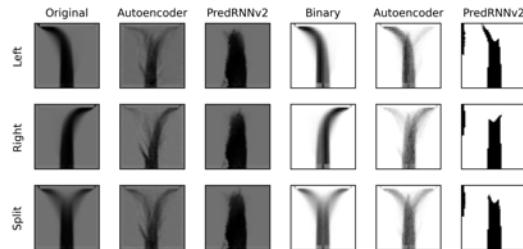


**Figure 7: Actual and Mean Predicted Channel Paths at step 90, with step 60 as the last information step**

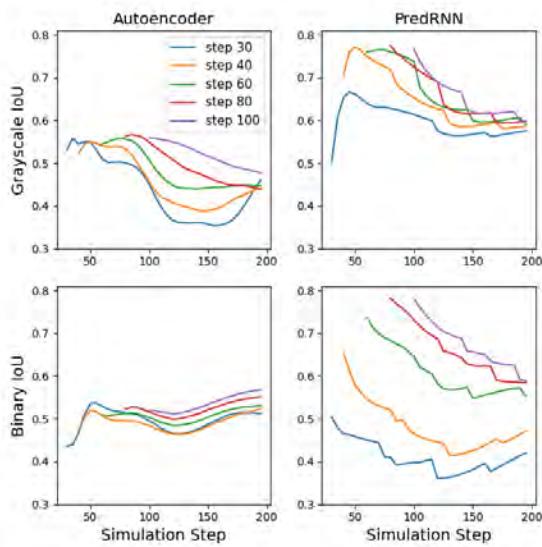
Figure 8 (step 200) highlights how the long-horizon biases of each model crystallise. In the grayscale domain, the autoencoder (second column) shows a Y-shape mean for every class, including those that should remain single-sided, revealing a tendency to oversplit. Its binary counterpart (fifth column), however, produces more similar means to the actual ones. By contrast, both PredRNN-V2 variants (third and sixth columns) converge on a thick, nearly vertical trunk, largely ignoring the desired left/right divergence and split, except for the binary **Left** class: that trunk tapers noticeably toward the upper left corner with a narrowing tip pointing in the correct direction.

Overall, the figure underscores two complementary trade-offs. Across architectures, autoencoders better preserve directional geometry but can introduce spurious splits in the grayscale setting, whereas PredRNN-V2 retains high mass overlap by defaulting to a thick central trunk, with only a tapered silhouette hinting at direction. Across input representations, grayscale frames provide context that helps both models recognise subtle bends yet also encourages the autoencoder to “hallucinate” extra branches; in contrast, binary frames strip away that context, enabling the autoencoder to stay truer to the actual left/right/split outcomes, while pushing PredRNN-V2 to an even more trunk-dominated, branch-poor forecast.

Figure 9 puts the qualitative trade-offs on a quantitative footing by plotting IoU as a function of forecast horizon for five “last-seen”



**Figure 8: Actual and Mean Predicted Channel Paths at step 200, with step 60 as the last information step.**



**Figure 9: IoU Curves of Path Prediction By Model and Image Type.**

input times (30–100). Two broad patterns emerge. Across models, PredRNN starts ahead but erodes faster: in both representations its earliest-start curves (blue/orange) begin 10–15 pp above the autoencoder, yet by 150–200 steps the gaps narrow or even invert, reflecting the thick-trunk bias that inflates early overlap but cannot track fine geometry at long range. Across representations, grayscale favours PredRNN while binary favours the autoencoder: PredRNN's grayscale IoU stays above 0.60 out to step 140, whereas its binary counterpart falls below 0.45; conversely, the autoencoder's binary IoU climbs steadily and overtakes its grayscale curve beyond step 120. Taken together, the curves confirm the visual impressions: PredRNN excels when rich intensity context is available and horizons are short, whereas the autoencoder is more resilient over long horizons—especially when only skeletal (binary) channel shapes are provided.

## 5 SUMMARY AND FUTURE WORK

**Final Direction Prediction.** Our experiments establish a clear hierarchy among the single-frame classifiers. On full-resolution

grey-scale images the hybrid Autoencoder–CatBoost (ACB) model outperforms a lightweight convolutional network by roughly four percentage points in the early-warning window (steps 40–80); its latent code appears to filter background noise while the boosted trees handle the small sample regime. Yet the ranking flips when inputs are aggressively simplified: the CNN becomes more sample-efficient on binarized or very low-resolution images, suggesting that architectural inductive bias can compensate for lost pixel fidelity. Resolution tests reveal graceful degradation down to  $10 \times 10$  but an abrupt drop at  $6 \times 6$  for CNN at the later stage.

Several research avenues could address these limitations and broaden the scope of the work. Class-balancing strategies—focal loss, re-weighting, or tailored augmentation—are natural first steps toward improving SPLIT recall. Exploring richer architectures such as attention mechanisms, transformers, or physics-guided networks may further reconcile global context with local texture. Complementary directions include uncertainty quantification for risk-aware decision support, self-supervised pre-training on unlabelled flow imagery, and alternative metrics that capture topological fidelity. Together, these extensions promise to strengthen channel-direction forecasting while opening the door to related problems in erosion and geomorphology.

**Path forecasting.** For long-horizon trajectory prediction, we contrasted a recurrent video predictor (PredRNN-V2) with a latent-evolution model that propagates auto-encoded states through an MLP. PredRNN-V2 attains the highest short-term intersection-over-union, chiefly by extending a thick, high-confidence “trunk,” yet its outputs blur and develop topological errors after roughly 100 simulation steps. The autoencoder-MLP, by contrast, preserves channel geometry much farther into the future—especially on binary inputs—though at the expense of lower early-step overlap. These complementary behaviours expose a fundamental trade-off between instantaneous accuracy and long-range fidelity.

Future work could hybridize the two approaches, for example, using PredRNN-V2 for the first few dozen frames before handing off to a geometry-aware latent evolution, integrate explicit level-set or signed-distance representations, and quantify predictive uncertainty so practitioners know when to refresh forecasts with new sensor data. Re-examining evaluation criteria is equally important: metrics that emphasise topology or centre-line accuracy may convey practical utility better than IoU alone. Finally, spatiotemporal-LSTM architectures such as the one explored by Lyu [13] merit re-assessment under a rigorously leak-free split and direct comparison with the PredRNN baseline reported here.

**Toward an integrated pipeline.** Bridging the two tasks offers additional gains: direction classifiers could provide coarse constraints that prune impossible trajectories, while path forecasts could generate self-supervised labels for frames where ground truth is scarce. An end-to-end, multi-resolution model that fuses raw imagery with hydraulic state variables, learns class-balanced objectives, and updates online as fresh measurements arrive may ultimately turn these proof-of-concept models into a practical early-warning tool for erosion control systems.

**[Acknowledgement.]** We thank Isaac Khor (a former Clark student) for his earlier work on related topics and thank Zhou Lyu and Dr. Chris H. Rycroft for stimulating discussions.

## REFERENCES

- [1] Alan D. Howard and Charles F. McLane. 1988. Erosion of cohesionless sediment by groundwater seepage, *Water Resources Research* **24**, 1659-1674 (1988). <https://doi.org/10.1029/WR024i010p01659>
- [2] Peter Dietrich. 2005. *Flow and transport in fractured porous media* (Springer, 2005).
- [3] Thomas Dunne. 1980. Formation and controls of channel networks. *Progress in Physical Geography* **4**, 211-239 (1980). <https://doi.org/10.1177/030913338000400204>
- [4] Robert L. Leheny and Sidney R. Nagel. 1993. Model for the evolution of river networks. *Physical Review Letters* **71**, 1470-1473 (1993). <https://doi.org/10.1177/030913338000400204>
- [5] Michael Berhanu, Alexander Petroff, Oliver Devauchelle, Arshad Kudrolli, and Daniel H. Rothman. 2012. Shape and dynamics of seepage erosion in a horizontal granular bed. *Physical Review E* **86**, 041304 (2012). <https://doi.org/10.1103/PhysRevE.86.041304>
- [6] Arshad Kudrolli and Xavier Clotet. 2016. Evolution of porosity and channelization of an erosive medium driven by fluid flow, *Physical Review Letters* **117**, 028001 (2016). <https://doi.org/10.1103/PhysRevLett.117.028001>
- [7] Arshad Kudrolli, Nikolay Ionkin, and Andreea Panaiteescu. 2017. Headward growth and branching in subterranean channels. *Physical Review E* **96**, 052904 (2017). <https://doi.org/10.1103/PhysRevE.96.052904>
- [8] Bjørnar Sandnes, Eirik G. Flekkøy, H.A. Knudsen, Knut J. Maloy and H. See. 2011. Patterns and flow in frictional fluid dynamics, *Nature Communications* **2**, 288 (2011). <https://doi.org/10.1038/ncomms1289>
- [9] Amala Mahadevan, Ashish V. Orpe, Arshad Kudrolli and Lakshminarayanan Mahadevan. 2012. Flow-induced channelization in a porous medium. *Europhysics Letters* **98**, 58003 (2012). <https://doi.org/10.1209/0295-5075/98/58003>
- [10] Chao Liu, Han Li, Jizhe Xu, Weijun Gao, Xiang Shen, and Sheng Miao. 2023. "Applying Convolutional Neural Network to Predict Soil Erosion: A Case Study of Coastal Areas", *International Journal of Environmental Research and Public Health*, **20** (3), (2023). <https://doi.org/10.3390/ijerph20032513>
- [11] Goldi Jarbais and Pon Harshavardhanan, "Identifying the primary drivers of soil erosion through machine learning: A comparative analysis of three algorithms Open Access", *Journal of Hydroinformatics* (2024) **26** (11): 2837-2851. <https://doi.org/10.2166/hydro.2024.181>.
- [12] Mostafa Delpisheh, Benyamin Ebrahimpour, Abolfazl Fattahi, Majid Siavashi, Hamed Mir Hosseini Mashhadimoslem, Mohammad Ali Abdol, Mina Ghorbani, Javad Shokri, Daniel Niblett, Khabat Khosravi, Shayan Rahimi, Seyed Mojtaba Ali-rahimi, Jiaoshui Yu, Ali Elkamel, ke Vahid Niasar and Mohamed Mamlouk. 2024. Leveraging machine learning in porous media. *Journal of Materials Chemistry A*, **12**(32), 20717-20782 (2024). <https://doi.org/10.1039/D4TA00251B>
- [13] Z. Lyu, Data-driven Analysis and Modeling of Erosion Networks. 2021. Master of Engineering thesis. Institute for Applied Computational Science, Harvard University, Cambridge, Massachusetts (2021).
- [14] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the Dimensionality of Data with Neural Networks. *Science* **313**, 504–507 (2006). <https://doi.org/10.1126/science.1127647>
- [15] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna V. Dorogush, and Andrey Gulin. 2017. CatBoost: unbiased boosting with categorical features. [arXiv:1706.09516 \[cs.LG\]](https://doi.org/10.48550/arXiv.1706.09516) (2017). <https://doi.org/10.48550/arXiv.1706.09516>
- [16] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. 2021. PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning. [arXiv:2103.09504 \[cs.LG\]](https://doi.org/10.48550/arXiv.2103.09504) (2021). <https://doi.org/10.48550/arXiv.2103.09504>
- [17] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. [arXiv:1506.04214 \[cs.CV\]](https://doi.org/10.48550/arXiv.1506.04214) (2015). <https://doi.org/10.48550/arXiv.1506.04214>

Received 15 May 2025