U-Net Segmentation for the Detection of Convective Cold Pools From Cloud and Rainfall Fields

Jannik Hoeller[1,2], Romain Fiévet[2], Jan O. Haerter[1,2,3]

[1]Leibniz Centre for Tropical Marine Research, Fahrenheitstr. 6, 28359 Bremen, Germany
[2]Niels Bohr Institute, Copenhagen University, Blegdamsvej 17, 2100 Copenhagen, Denmark
[3]Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany

Corresponding author: Jannik Höller, jannik.hoeller@leibniz-zmt.de

Abstract
Convective cold pools (CPs) are known to mediate the interaction between convective rain cells and thereby help organize thunderstorm clusters, in particular mesoscale convective systems and extreme rainfall events. Unfortunately, the observational detection of CPs on a large scale has so far been hampered by the lack of relevant large-scale near-surface data. Unlike numerical studies, where high-resolution near-surface fields of relevant quantities such as virtual temperature and winds are available and frequently used to detect cold pools, observational studies mainly identify CPs based on surface time series. Since research vessels or weather stations measure these time series locally, the characterization of cold pools from observations is limited to regional or station-based studies. To eventually enable studies on a global scale, we here develop and evaluate a methodology for the detection of CPs that relies only on data that (i) is globally available and (ii) has high spatio-temporal resolution. We trained convolutional neural networks to segment CPs in cloud and rainfall fields from high-resolution cloud resolving simulation output. Such data is not only available from simulations, but also from geostationary satellites that fulfill both (i) and (ii). The networks make use of a U-Net architecture, a common choice for image segmentation due to its strength in learning spatial correlations at different scales. Based on cloud and rainfall fields only, the trained networks systematically identify CP pixels in the simulation output. Our methodology may thus open for reliable global CP detection from space-borne sensors. As it also provides information on the spatial extent and the relative positioning of CPs over time, our method may offer new insight into the role of CPs in convective organization.

1 Introduction

CPs are volumes of atmospheric air that are cooled by the evaporation of precipitation. The resultant body of air is denser than its surroundings and therefore experiences negative buoyancy (Markowski & Richardson, 2011), giving rise to a so-called convective downdraft, or microburst (Lundgren et al., 1992). When reaching earth's surface, CPs spread horizontally as density currents (Droegemeier & Wilhelmson, 1985; Zuidema et al., 2017; Drager & van den Heever, 2017). While expanding radially along the surface, the CP can be characterized as consisting of (i) a deeper head, which can measure between hundreds of meters and several kilometers vertically (Droegemeier & Wilhelmson, 1985) (Benjamin, 1968); and (ii) a shallower interior, which is separated from the head by a wake region (Benjamin, 1968; Droegemeier & Wilhelmson, 1987).

Substantial mechanistic significance has been attributed to the thin surface of horizontal convergence between the CP head and the ambient atmosphere. On the one hand, this region, which is often referred to as the CP gust front typically features pronounced vertical wind speed. On the other, the initial negative buoyancy anomaly near the CP's gust front is gradually reduced as the CP spreads, a consequence of enhanced surface latent and sensible heat fluxes (Tompkins, 2001; Torri & Kuang, 2016; Drager et al., 2020). Thus, in the course of the lateral expansion, warm ambient air can be lifted upwards (Drager & van den Heever, 2017), and further condensation and convection can result. Both the mechanical and thermodynamic effects at the gust front can thus encode a mechanism for "communication" between current and future precipitation cells (Simpson, 1980).

Although the relative contributions of thermodynamic and dynamical mechanisms, by which CPs can trigger new convection, are still under debate (Torri et al., 2015; Romps & Jeevanjee, 2016; Fuglestvedt & Haerter, 2020) and do depend on the specific case (Feng et al., 2015), there is substantial consensus that CPs are able to trigger new clouds in their vicinity and modify the subcloud moisture distribution (Böing et al., 2012; Schlemmer & Hohenegger, 2016; Drager & van den Heever, 2017). Interaction mechanisms, mediated through CPs, have inspired a number of conceptual studies, aiming to mimic emergent self-organization by assuming simple, yet plausible processes (Böing, 2016; Haerter

et al., 2019; Haerter, 2019; Nissen & Haerter, 2021). In recent idealized cloud-resolving simulations, CPs were indeed shown to promote clustering processes such as seen in mesoscale convective systems (MCS) (Haerter et al., 2020; Jensen et al., 2021). Observationally, MCS have been implicated in the majority of tropical extreme rainfall events (Tan et al., 2015) and may play a role in future intensification of extremes (Tan et al., 2015; Fowler et al., 2021).

Acknowledging the importance of CPs for weather prediction and climate, several CP characteristics have been investigated over the past decades, most of them within numerical studies: virtual temperature anomalies have been used to track CPs in cloud-resolving simulations by detecting contiguous patches (Schlemmer & Hohenegger, 2016) or by employing unsupervised image segmentation (Gentine et al., 2016). Drager and van den Heever (Drager & van den Heever, 2017) compared the utility of different variables for CP identification in numerical model output. Using the virtual temperature gradient their tracking method allowed for the study of average CP properties. Torri and Kuang (Torri & Kuang, 2019) used a Lagrangian tracking algorithm to investigate CP collisions and their impact on CP life and dynamics in Eulerian and Lagrangian models. Focusing on the dynamical gust front, Fournier and Haerter (2019) and Henneberg, Meyer, and Haerter (2020) introduced tracking algorithms targeting the thin convergence rings surrounding each CP, respectively exploiting radial velocity gradients and tracer particles emitted at the perimeter of precipitation patches.

Simulations carry the advantage of detailed analysis of specific mechanisms and offer essentially continuous output data for many variables, available over the entire model domain. This accessibility simplifies the detection and tracking of CPs in the model domain and allows for profound investigations of CP characteristics. Yet, even under advances in computing capabilities, numerical studies still depend on the model chosen, so that immediate conclusions with respect to the real world remain ambiguous. In this regard a key limitation is posed by the required model resolution: since traditional general circulation models (GCM) are too coarse to resolve CP processes (Feng et al., 2015; Fiévet et al., 2022), CP mechanisms are mostly studied in high resolution simulations within limited domain sizes or, less often, by including specific parameterizations (Grandpeix & Lafore, 2010; Rio et al., 2009). In both cases, the validity of the outcome is limited by artificially imposed model constraints.

Exploiting the benefits of both, a simulation-observation hybrid was employed by Feng et al. (2015), who paired a high resolution regional model with month-long observational data from the AMIE/DYNAMO field campaign conducted over the tropical Indian Ocean. They identified CPs subjectively by spotting boundaries of echo-free regions in radar measurements and manually tracing them back to precipitation events. In the simulation, CPs were first detected by buoyancy thresholds (Tompkins, 2001), as detailed above. Isolated and intersecting CPs were distinguished via a watershed technique to be able to investigate mechanisms of convective cloud organization by CPs. Another approach to identify CPs in observations was implemented by Redl, Fink, and Knippertz (Redl et al., 2015), who developed a CP detection method based on surface observations from a network of weather stations located south of the Atlas Mountains in Morocco and Algeria, as well as satellite microwave data.

Observational data offers the possibility to investigate CPs under realistic conditions and enables the validation of results obtained from numerical studies. However, CP detection based on observational data is still challenging and the few methods which have been used so far did always employ some kind of ground-based measurement (e.g. radar measurements or a network of weather stations) to detect CPs in the local measurement area. The main reason for this is based on the fact that CPs form below precipitating clouds and spread just above the earth's surface, which complicates their detection from space. Unfortunately, due to the limited spatiotemporal coverage and accessibility of the employed data, these methods are not suited for comprehensive investigations on a global

scale, which are needed to derive conclusions about the role of CPs and convective organization with respect to climate change.

For this reason, we developed a CP detection algorithm based on convolutional neural networks which relies only on quantities that are observable from geostationary satellite imagery and thus exhibit a good spatiotemporal coverage. The algorithm was trained and tested with data from an idealized cloud-resolving simulation, where all field variables are available over the entire domain. To our knowledge, this is the first approach for CP detection which can be applicable to observational data on a global scale. Our algorithm may allow for new insight on the role of CPs in convective organization and the formation of weather extremes. Real-world CP detection results may further serve as a benchmark for CP representation in numerical models for weather and climate.

2 Methods

Convolutional neural networks are widely used in classification and segmentation problems, where a field of input data is gradually coarsened through a filtering operation. Upon each filtering step, spatial correlations at larger and larger scales are distinguished. Whereas classification algorithms group the entire input data field into a set of classifiers, segmentation returns to the resolution of the input to mark each pixel as being of one of several categories. For the problem at hand, we wish to mark each pixel in the 2D plane as either belonging to a CP or not — thus the segmentation technique is appropriate.

2.1 Simulation Data

In order to simplify the generation of labeled data sets, the network training and testing is conducted using data from numerical simulations. To this end, the cloud-resolving three-dimensional atmosphere simulator System for Atmospheric Modeling (SAM) (Khairoutdinov & Randall, 2003), version 6.11, is used. It resolves the Navier-Stokes equations in the anelastic approximation on a staggered mesh. Convective fluxes are evaluated using a fifth-order finite difference scheme from Yamaguchi, Randall, and Khairoutdinov (2011) and turbulent dissipation is modeled by an eddy-viscosity based closure. Moist thermodynamics is resolved by transporting liquid and ice water static energy, total precipitating and non-precipitating water mass fractions, and uses a bulk single-moment microphysics closure scheme.

The configuration chosen for this study corresponds to an atmosphere over an idealized moist tropical land surface. It is similar to the configuration studied by Jensen et al. (Jensen et al., 2021) which exhibited strong and complex CP activity, and is therefore suited to design and test our detection method. The computational domain has a size of $L_x = L_y = 240$ km in the horizontal directions, and extends vertically to a maximum altitude of L_z of 26 km. It is discretized by an orthogonal mesh of horizontal resolution $\Delta x = \Delta y = 200$ m and vertical resolution Δz increasing from $\Delta z(z = 25\text{m}) = 50$ to $\Delta z(z = 25\text{km}) = 1000$ m over 100 levels. In the following, we use $n_x, n_y \in [0, N[$, with $N = 1200$ the linear horizontal domain size, as integers labeling the indices of the horizontal model grid. The lateral boundary conditions are set to be periodic. Relevant two-dimensional simulated fields are sampled instantaneously every 10 min. We refer to this as the "time step" throughout the paper.

Surface heat fluxes are evaluated using Monin-Obukhov similarity theory with a saturated humidity (moist ground condition) and a prescribed diurnally-varying temperature T, with an average of $T_0 = 298$ K. Its amplitude ΔT is chosen to represent plausible ranges measured for tropical land (Sharifnezhadazizi et al., 2019). The effect of the surface forcing is to trigger idealized diurnally-varying convective activity typical of tropical land surfaces: moist convection tends to develop during the afternoon hours

and MCS self-organize — giving rise to a complex organizational pattern. The nocturnal cooling then reduces convective activity and precipitation rates typically reach a domain-wide minimum during the early morning hours of the subsequent model day. In order to work with a diverse set of atmospheric conditions, four different configurations are run (Tab. 1), where $\Delta T \in \{2, 4\}$ K and wind shear is either switched off or set to a temporally and spatially averaged vertical profile over the trade wind regions (LAT: 5.5° to 16° N, LON: -20° to 10°) obtained from ERA5-measurements in July 2016. The vertical profile consists in a piece-wise linear profile with zero velocity below $z = 1$ km, linearly-increasing speeds from 0 to 16 m/s up to 19km-altitude and 16 m/s beyond.

Run	**Simulation**	ΔT [K]	**Wind shear**
1	**diu2K**	2	No
2	**diu2Kwind**	2	Yes
3	**diu4K**	4	No
4	**diu4Kwind**	4	Yes

Table 1: **Configuration of all numerical simulations.**

Ground Truth Labeling

Labeled data sets are derived from simulation output based on a CP detection and tracking algorithm (CoolDeTA). We employ the k-means algorithm to determine pixel-wise potential CP areas without defining a fixed threshold, along with a watershed algorithm. The starting points for the watershed filling are all locations with surface rain intensity, r_{int}, exceeding a threshold of $r_{int} \geq 2\ mm\,h^{-1}$. Providing the fields of virtual temperature, T_v, and both horizontal and vertical wind speed in the lowest domain level, as well as r_{int}, CoolDeTA identifies and tracks each CP instance individually and stores additional information, such as its age, i.e. the number of time steps since it was first detected by CoolDeTA. To use the simplest possible case, the derived labels for the present study are kept binary, comprising the two classes "CP" and "no CP."

Input Variables

Regarding the potential input for the neural network, SAM outputs several variables which are accessible from space-borne data. The present study focuses on the cloud top temperature, T_B, which is equivalent to the brightness temperature commonly used in remote sensing, and r_{int}. These two quantities are readily available from infrared emissions and an increasing number of precipitation products. Depending on the region of interest and problem specific requirements in terms of spatial and temporal resolution, as well as accuracy, these can be multi-satellite products with global coverage such as IMERG (Huffman et al., 2015) or even products based on ground-based weather radars.

Training and Test Sets

For each of the four simulation setups, output is available for 7.5 simulation days in total. After three days of spin-up, we employ the simulation output of day four for network training (75%) and validation (25%). As common in supervised learning, we randomly split the data by assigning each instance with a probability of 75% to the training set and with 25% to the validation set. While we train networks based on the training set, the validation set is used to monitor the progress of the training on separate data

which has not been trained on. Accordingly, we keep the obtained allocation fixed during the whole training and - in order to facilitate the comparison between networks - also during different network trainings.

When all network trainings are completed, the final performance is evaluated based on a test set (Willemink et al., 2020) consisting of simulation output of day six, i.e., the test set is not considered at any earlier stage. Although the observed CPs, and the complex pattern formed by their interaction, are unique for each simulation day, the offset of one day between the test set and the training and validation set guarantees fully independent sets w.r.t. the distribution of relevant quantities such as humidity.

To ensure sufficient variation between consecutive time steps of the data sets, we consider only every second time step of the corresponding simulation output for the training and validation set, and every fourth time step for the test set. This is particularly important for the test set to prevent any distortion of the final results due to correlated data.

In order to reduce the computational cost and accelerate network training, we subdivided every $N \times N$ pixel output field, termed "image," into 100 sub-regions, which we refer to as "patches," of $n_p \times n_p$ pixels each. As our downsampling requires the integer n_p to be a power of two (*see* Sec. 2.2) and to compromise between computational effort and prediction skill, we chose $n_p = 128$ as the linear dimension of each two-dimensional patch. To accomplish this, each original output image is padded from $n_x = 1200$ to $n_{x,pad} \equiv 1280$ in "wrap" mode, i.e., assuming a horizontally repeated model domain which ensures consistency with the periodic lateral boundary conditions.

Eventually, each network prediction requires an input and a corresponding ground truth to optimize and/or evaluate the network performance. While the ground truth corresponds to an $n_p \times n_p$ pixel patch of the output images with derived labels, the input consists of stacked patches corresponding to T_B and r_{int}. To compensate for lacking context information at patch boundaries, the input towers above the boundaries of the underlying ground truth patch by $n_p/2$ pixels on either side, resulting in an input patch size of $2n_p \times 2n_p$ pixels (Fig. 1). Although the additional $n_p/2$ pixels on each side are thus ranging into the adjacent ground truth patch, this overlap does not distort the results as the final network prediction only comprises the underlying central $n_p \times n_p$ pixel patch.

To ensure a robust training process and reliable results, we manually checked the ground truth labeling of every patch in the data set. We omitted patches if they (i) contained at least one, but less than 25 pixels (i.e., 1 km^2) of class "CP," (ii) were in the center of a large MCS with a gust front significantly beyond the boundaries of the input patch, (iii) were poorly labeled by the CP detection algorithm or (iv) featured ambiguous scenes where an unequivocal verification of the labeling is not possible. For the evaluation of both (iii) and (iv) the dynamical gust front, i.e., $w > \overline{w} + 2\sigma_w$ served as main indicator: clear offsets between gust front and boundaries of ground truth CPs were interpreted as poor labeling, discontinuous and thus dissipating gust fronts as ambiguous cases.

As simulation setups affect the cloud and rainfall patterns associated with CPs, we considered patches from simulations with different environmental conditions. Yet, both simulations with imposed wind profile feature prevailing easterly winds. To allow the network to capture underlying patterns independent of the wind direction, we rotate each patch of the two simulations with wind by 90°, 180° and 270° and add the resulting patches to the data sets. Extending data sets with slightly modified copies of the data based on operations such as rotations or translations is a common approach to increase the amount and diversity of data and is called data augmentation.

Data imbalances due to the under-representation of classes or features in the training set are a common issue of learning algorithms (He & Garcia, 2009). Taking the reduced convective activity due to nocturnal cooling into account, the majority of patches do not contain any CP pixels for the ground truth-labeled data and features only the class "no CP." We compensated for this by randomly removing a certain number of these patches (Shi et al., 2021). By experiment, we selected the number of patches with only class "no CP" to be 4% of the training and validation set. The other extreme are patches with class "CP" only. It is known that surface temperature oscillations promote the sudden organization of CPs into MCS (Haerter et al., 2020; Jensen et al., 2021). Since the surface areas of these MCS often exceed the patch size, a great number of patches has class "CP" only. However, omitting patches in the center of large MCS according to (ii) already lowered the number of patches with class "CP" only to $\approx 5.5\%$ resulting in a sufficiently balanced training and validation set distribution (Fig. 2). We chose not to balance the distribution of the test set in order to not affect the results in any way.

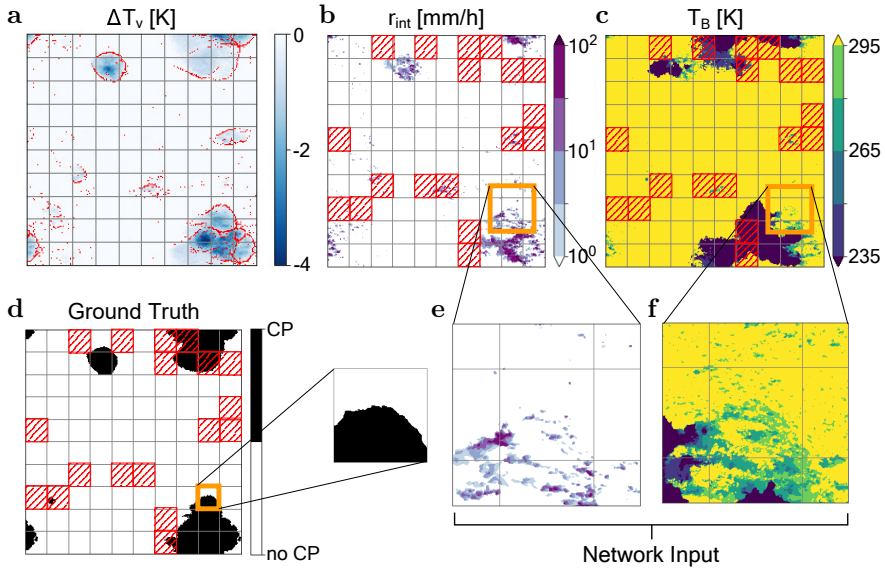

Figure 1: **Defining patches for neural network input and ground truth. a**, Time step 497, i.e., 80 min before T_{max} on simulation day 4, of "diu4K wind", showing near-surface virtual temperature anomaly, ΔT_v, with superimposed dynamical gust front, i.e., $w > \overline{w} + 2\sigma_w$ (red scatter); The superimposed grid represents the individual $n_p \times n_p$ pixel patches, processed by the neural network; **b**, Analogous to (a) but for surface rain intensity, r_{int}. Patches that were omitted from the data set are hatched; **c**, Analogous to (a) but for cloud top temperature, T_B. **d**, Ground truth labeling showing CP areas as black regions; a single patch is enlarged for clarity; **e**, Highlighted patch, including padding, for r_{int}; **f**, Analogous to (e) but for T_B.

2.2 Network Architecture

As mentioned, instead of predicting one specific label per provided input image (classification), the detection of CPs requires an output, such as "CP" or "no CP," for every pixel of the image (segmentation). A common architecture used for segmentation is the U-Net (Ronneberger et al., 2015), a convolutional neural network (CNN) that consists of an encoder path and a decoder path. In the encoder path input images are downsampled after every block, allowing the network to learn features at larger scales. A common downsampling method where the output is generated from the input by considering only the maximum value of a moving window of size $s \times s$ and which we also ap-

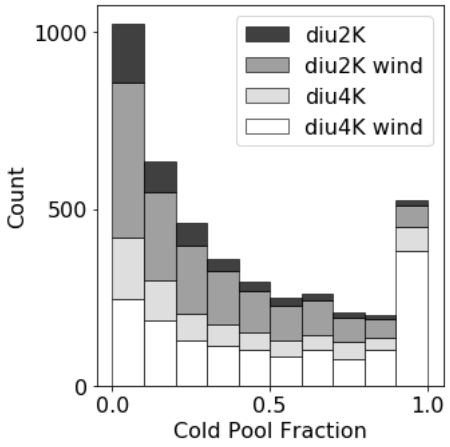

Figure 2: **Training and validation set distribution with respect to the fraction of class "CP" in each patch.**

ply in the present study with $s = 2$, is max pooling. By reducing the resolution of the image in each downsampling step, typically by a factor of two as we do here, the network can learn features at different scales. To be able to capture the underlying correlations, the number of filter layers is doubled with every downsampling step. In the decoder path, on the other hand, the images are upsampled again via transposed convolution or interpolation to finally enable pixel-wise predictions. After each upsampling step, concatenated filter layers of the same depth encoder block provide additional information. The employed U-Net architecture for the simplified case with three vertical blocks ($n_b = 3$) is depicted in Fig. 3.

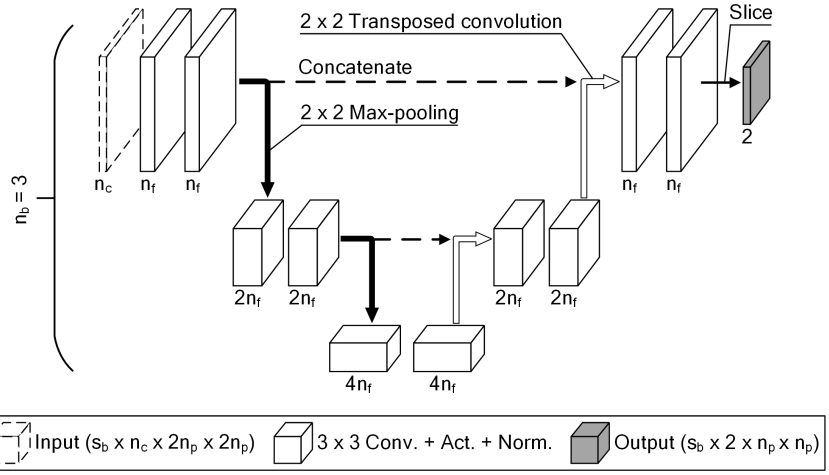

Figure 3: **U-Net architecture for cold pool segmentation.** The schematic shows the case with three filtering blocks ($n_b = 3$). The number of input channels n_c represents the number of different variables provided to the network as input. In the case of pseudo-3D models the number of input channels, n_c = number of variables x number of utilized time steps, n_t. The number of output channels comprises the two classes "CP" and "no CP".

Apart from n_b and the starting number of filter layers n_f, neural networks and U-Nets in particular offer a variety of modeling choices, termed hyperparameters, to tune.

After an exploration phase, in which we identified hyperparameters significant for our network along with promising orders of magnitude based on training and validation performances, we investigated the following seven hyperparameters in more detail: n_b, ultimately chosen as $n_b = 6$; n_f, ultimately chosen as $n_f = 64$; the activation function, ultimately chosen as LeakyReLU; the normalization strategy, ultimately chosen as batch normalization; the loss function, ultimately chosen as combination of cross entropy loss and dice loss; the learning rate l_r, ultimately chosen as exponentially decaying function $l_r = 10^{-5} \times \gamma^{e_t}$ with e_t as the training epoch and $\gamma = 0.9$; and the batch size s_b, ultimately chosen as $s_b = 8$. Activation functions are nonlinear functions and a fundamental part of CNNs. Following convolutional layers in the convolution block (cf. Fig. 3), activation function enable the network to capture complex patterns. Typically, convolution blocks are completed by normalization steps, which can support an efficient learning process (Ioffe & Szegedy, 2015). While the loss function is the function to be minimized during training, l_r controls the corresponding optimization step size. The number of instances considered per optimization step is the batch size. Typically, training batch sizes are greater one to reduce the risk of getting stuck in local minima.

In order to determine the most promising network configuration w.r.t. the seven hyperparameters, we conducted a number of experiments based on the training and validation set. Instead of analyzing all possible combinations of configurations, we limited the number of experiments by structuring them in two stages. Starting from a first guess reference configuration for which all seven hyperparameters were defined pragmatically, the first stage consists of multiple levels, each containing experiments for a group of hyperparameters with all their combinations. After each level, the reference configuration is updated based on the best candidates of those hyperparameters. Due to their close relation, we grouped l_r with s_b (group 1), activation function with normalization strategy and loss function (group 2), and n_b with n_f (group 3). Whereas the hyperparameters in group 1 are essential for robust learning and thus investigated first, the hyperparameters in group 3 are examined last as larger numbers of n_b and n_f, which were expected to be advantageous, would slow down the remaining experiments significantly.

Since some hyperparameters could have candidates with similarly good performance so that the best candidate might thus change for other configurations, we performed a second stage of experiments with all combinations of these candidates plus some fine-tuned ones.

Depending on the convolution kernel, CNNs can be categorized into 2D and 3D CNNs. Conventional end-to-end 2D CNNs receive 2D input, which may consist of multiple channels, e.g. 2D fields of different variables, apply 2D convolutions, i.e., convolutions with 2D kernel matrix, and generate a corresponding 2D output, whereas 3D CNNs analogously process 3D data. At the expense of significantly higher computational cost, 3D CNNs are thus able to learn correlations in a third dimension based on the 3D convolution kernel. As we are interested in 2D segmentations and the simplest model possible, we selected the 2D version. However, since CPs are density currents and exhibit gust fronts typically emanating radially from a precipitation cell center, expansion over time constitutes one of the main CP features (Benjamin, 1968). In order to include this time-dependent component and potentially enable the network to learn the correlations between consecutive time steps, we also implement the so-called pseudo-3D approach. The term "pseudo-3D", introduced by Vu, Grimbergen, Nyholm, and Löfstedt (2020), represents a model class that is intermediate between conventional 2D CNNs and 3D CNNs. In pseudo-3D models the information of the third dimension (here time) is inserted as additional input channels to the network, therefore without modifying the network's 2D architecture. As a consequence, the total number of input channels of pseudo-3D models depends not only on the number of input variables provided, but on the product of the numbers of input variables and utilized time steps. Thus, pseudo-3D models might potentially benefit from time-dependent information without being as computationally

expensive as end-to-end 3D models (Vu et al., 2020). In the present study, we investigate the pseudo-3D model with three (p3D3t) and five time steps (p3D5t). Time steps are thereby centered about the time step for which a prediction is to be made.

2.3 Loss and Evaluation Metrics

The selection of an appropriate loss function depends on the specific problem at hand. All loss functions use the pixel-wise network prediction $U = [U_0, U_1]$, consisting of the two output channels U_0, $U_1 \in \mathbb{R}^{n_p \times n_p}$, that is,

$$U^{(0)} = \begin{bmatrix} {U^{(0)}}_{11} & \dots & {U^{(0)}}_{1n_p} \\ \vdots & \ddots & \vdots \\ {U^{(0)}}_{n_p1} & \dots & {U^{(0)}}_{n_pn_p} \end{bmatrix}, U^{(1)} = \begin{bmatrix} {U^{(1)}}_{11} & \dots & {U^{(1)}}_{1n_p} \\ \vdots & \ddots & \vdots \\ {U^{(1)}}_{n_p1} & \dots & {U^{(1)}}_{n_pn_p} \end{bmatrix}, \tag{1}$$

where indexes "0" and "1" indicate the "no CP" and "CP" channels, respectively, and compare U with the corresponding ground truth, denoted $V \in \mathbb{N}^{n_p \times n_p}$, where $V_{ij} \in \{0, 1\}$, indicating "no CP" and "CP," respectively. V is derived by CoolDeTA as

$$V = \begin{bmatrix} V_{11} & \dots & V_{1n_p} \\ \vdots & \ddots & \vdots \\ V_{n_p1} & \dots & V_{n_pn_p} \end{bmatrix}. \tag{2}$$

We examined several loss functions during the experiments. For this purpose, we rescaled each pixel $U_{kl}^{(j)}$ in U to the range [0,1] so that the "probabilities" of both the "no CP" and "CP" channel sum up to one. We term the result of this so called "softmax" function u. The corresponding function is written as

$$u_{kl}^{(j)} \equiv \frac{e^{U_{kl}^{(j)}}}{e^{U_{kl}^{(0)}} + e^{U_{kl}^{(1)}}}, \text{for } j \in \{0, 1\}. \tag{3}$$

In order to compare u to the ground truth, we split V analogously to the prediction via one-hot encoding into two slices of binary data $v = [v^{(0)}, v^{(1)}]$ with

$$v^{(0)} = \begin{bmatrix} {v^{(0)}}_{11} & \dots & {v^{(0)}}_{1n_p} \\ \vdots & \ddots & \vdots \\ {v^{(0)}}_{n_p1} & \dots & {v^{(0)}}_{n_pn_p} \end{bmatrix} v^{(1)} = \begin{bmatrix} {v^{(1)}}_{11} & \dots & {v^{(1)}}_{1n_p} \\ \vdots & \ddots & \vdots \\ {v^{(1)}}_{n_p1} & \dots & {v^{(1)}}_{n_pn_p} \end{bmatrix}, \tag{4}$$

that is, $v_{kl}^{(0)} = 1 - V_{kl}$ and $v_{kl}^{(1)} = V_{kl}$. As loss functions we employed a cross entropy loss which is often used as default in image segmentation and defined as

$$\mathcal{L}_{CE}(u, v) = \sum_{j,k,l} \frac{-v_{kl}^{(j)} \log(u_{kl}^{(j)})}{\sum_{m,n,q} v_{nq}^{(m)}}, \tag{5}$$

a soft Dice coefficient loss, defined as

$$\mathcal{L}_{Dice}(u, v) = 1 - \frac{2 \sum_{j,k,l} u_{kl}^{(j)} v_{kl}^{(j)} + \epsilon}{\sum_{j,k,l} u_{kl}^{(j)} + \sum_{j,k,l} v_{kl}^{(j)} + \epsilon}, \tag{6}$$

where $\epsilon = 1$ is a constant preventing divisions by zero (Jadon, 2020), and a combination of both

$$\mathcal{L}(u, v) = \alpha \mathcal{L}_{Dice}(u, v) + (1 - \alpha) \mathcal{L}_{CE}(u, v), \tag{7}$$

with $\alpha = 0.5$. Whereas $\mathcal{L}_{Dice}$ can deal with imbalanced data sets (Milletari et al., 2016) and focuses on how good the predicted CPs overlap the ground truth CPs, $\mathcal{L}_{CE}$ evaluates the difference between the probability distributions of u and v. For our problem

we chose $\mathcal{L}$ as loss function as it combines the strengths of both $\mathcal{L}_{Dice}$ and $\mathcal{L}_{CE}$, and outperformed both these functions during the experiments.

For the evaluation of the trained networks we distinguish between patches containing only one of the two classes for the corresponding ground truth data and patches with at least one pixel of both classes. In the former case, the only evaluation metric will be pixel accuracy, PA, which evaluates the fraction of predictions that are correct, defined as

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \,. \tag{8}$$

In Eq. 8 TP and TN indicate true positive and true negative predictions, respectively, whereas FP and FN denote false positive and false negative predictions, respectively.

In case the ground truth patch contains at least one pixel of both classes, we additionally calculate the intersection over union, IOU,

$$IOU = \frac{TP}{TP + FP + FN} \,. \tag{9}$$

The IOU score is a measure of how well the specific objects of prediction and ground truth overlap one another, ranging from zero, where no overlap is found, to unity, for perfect overlap. Furthermore, we consider Precision and Recall, defined as

$$Precision = \frac{TP}{TP + FP} \,, \tag{10}$$

and

$$Recall = \frac{TP}{TP + FN} \,. \tag{11}$$

As IOU both Precision and Recall range from zero, where no "CP" pixel was correctly identified, to unity, for a perfect prediction. However, shedding light on different components of the prediction, they help to understand potential sources of good and bad performances.

2.4 Network Validation

We plot the training and validation losses for the 2D and both pseudo-3D models as a function of the epoch, e_t (Fig. 4). e_t describes how many times the entire training set has been passed through the neural network. The loss measures the quality of the prediction, where a value of zero means perfect prediction. Instead of defining a fixed e_t, we stop the training if the validation loss has not improved for ten consecutive e_t. Taking into account the stochasticity involved in the training process, we conducted three runs for each model. As might be expected, the *training loss* decreases monotonically with the data employed for learning, i.e., e_t, and reaches a value close to zero for our maximum e_t of 22—24. Notably, for intermediate e_t both pseudo-3D neural networks perform better than the 2D counterpart, whereas for the final e_t the three are essentially indistinguishable.

However, a good value of training loss does not necessarily imply optimal *validation loss*, a measure of prediction quality for a previously unseen data set. Indeed, we find that intermediate e_t (≈ 10) yield lowest validation loss for all three cases, such that a global mimimum occurs. This type of optimum at intermediate e_t is typical of neural networks and is often interpreted as large e_t constituting a form of overfitting w.r.t. the training data — yielding less than optimal behavior for the unknown validation data. Yet, the minimum is characterized by an asymmetric increase of validation loss, where somewhat larger e_t lead to only small increases in validation loss. Further, we again find quantitative improvements in validation loss for the pseudo-3D cases, which systematically reach lower values of loss than 2D.

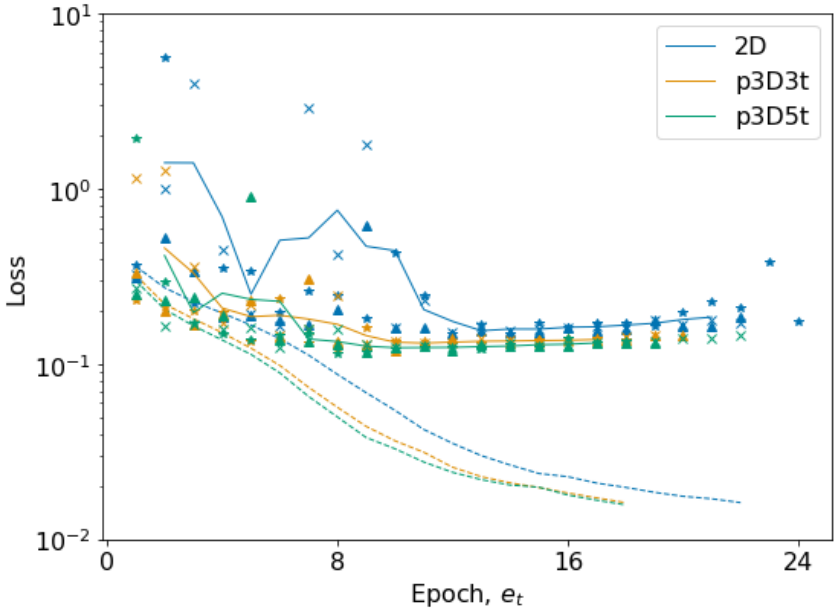

Figure 4: **Training and validation loss for different models.** Loss as a function of the epoch, e_t, for the 2D, p3D3t, and p3D5t neural networks. Dashed lines represent running averages of training loss for all training runs of a respective neural network type. Thin colored are running averages of validation loss for all training runs of a respective neural network type additionally averaged over a centered window of three e_t; different symbols correspond to the validation loss of the different training runs. Note: As the mean variance of the training loss for the three neural network types is only between 2.5×10^{-6} (2D) and 6.1×10^{-6} (p3D3t), markers for the training loss of different training runs are not visualized.

3 Results

For the final evaluation of the trained neural networks, we now employ the test set, i.e. day 6 of each simulation. We ensure that the results obtained are on the conservative side, by considering only the run with the greatest final validation loss for each model, i.e., the third run for the 2D model and the first run for both p3D3t and p3D5t (cf. Tab. 2).

We quantify the utility of our segmentation method by applying typical performance metrics (Tab. 3). A key measure is pixel accuracy (PA), which is generally high (mean PA$\gtrsim$ 94%) for all models, with the pseudo-3D models performing slightly better than the 2D model. The intersection over union (IOU) score denotes the fidelity of spatial overlap of ground truth CPs and neural network-predicted CPs, and is thereby sensitive to the underlying CP areas, yielding lower values than PA for all models. Again the pseudo-3D models achieve higher mean IOU scores of 0.75 (p3D3t) and 0.74 (p3D5t) compared with 0.71 for the 2D model. As mean Precision is almost equally high for all models (Tab. 3), the difference in IOU is mainly driven by the higher mean Recall of the pseudo-3D models, i.e., they miss less "CP" pixels than the 2D model.

In order to investigate the sensitivity of the network performances w.r.t. the CP fraction in the patch, we group PA and IOU score into quartiles of CP fraction. For all these quartiles PA is high (PA$\gtrsim$ .95) for all models (Fig. 5a). Yet, systematic differences exist: Generally, PA is greatest for small CP fraction and somewhat decreases for intermediate fractions, where it then seems to saturate. This behaviour is expected, since (i) the majority of the training and validation set patches contained only small fractions of

Model	Run	Final e_t	Final Loss	Mean Loss	Final PA [%]	Mean PA [%]
2D	1	22	0.173	0.166	93.6	93.5
	2	24	0.177	0.203	93.8	92.5
	3	22	0.186	0.163	93.5	93.6
p3D3t	1	19	0.147	0.137	94.4	94.3
	2	18	0.139	0.138	94.6	94.2
	3	20	0.146	0.138	94.5	94.4
p3D5t	1	22	0.148	0.134	94.6	94.7
	2	18	0.136	0.127	94.7	94.6
	3	19	0.133	0.128	94.8	94.7

Table 2: **Validation loss and pixel accuracy for the training runs of different models.** Columns indicate three runs for each model along with the final epoch, e_t, the final validation loss (final e_t), the mean validation loss, averaged over the final ten e_t, and final and mean pixel accuracy, PA.

class "CP", slightly biasing the neural networks towards "no CP" predictions and (ii) regions without "CP" pixels often feature neither precipitation, nor clouds, simplifying the network prediction. Overall, PA is somewhat greater for the pseudo-3D cases, however, this benefit is nearly lost for small CP fractions, a finding we attribute to the potential noise at the early stages of CP expansion: in p3D3t and p3D5t, where additional time steps are included, data taken before the onset of the CP might contribute to the training — thus obscuring the signal of actual CP expansion.

The IOU score (Fig. 5b) can be substantially lower for the smallest CP fraction quartile, with some improvement for the pseudo-3D models. This loss for small CP fraction is however not surprising to us, as for small CP fraction there will often be only few pixels in a patch which actually qualify as CP pixels and small spatial displacements of these pixels in the predicted data can already lead to a drastic reduction of the IOU. Refined measures could be designed that still assign a score to a minimally displaced CP pixel. However, physically relevant CPs, e.g., in terms of collision effects (Meyer & Haerter, 2020; Fiévet et al., 2022) and intense precipitation (Jensen et al., 2021) tend to cover larger patch fractions and the IOU score is systematically high — again with best performances for the pseudo-3D cases.

We now turn to test patches which contain only "no CP" or "CP" pixels in the ground truth. For the former case, PA yields near-perfect accuracy (Tab. 4). Thus, the models shows high fidelity in capturing cases where CPs are not present, most likely due to the absence of precipitating clouds in a majority of the patches. PA is however substantially reduced in the latter case (Tab. 5). The reduction in PA is especially pronounced for p3D5t, thus the model where five time steps were used. We attribute this loss of accuracy to the temporal mixing of patches with and without CP pixels, whereby the lack of CP pixels at earlier stages may skew the results.

To enable a more application-oriented perspective on the performance of the three models, we evaluate the percentage of successfully detected CPs as a function of CP area (Fig. 6). For this purpose, we define CPs as spatially 4-connected regions of ≥ 25 "CP" pixels ($\geq 1\ km^2$). The minimum CP size of 25 "CP" pixels ensures that only robust predictions are considered. A ground truth CP is considered detected if a predicted CP overlaps more than 50% of its area and if more than 50% of the area of the predicted CP falls inside the ground truth CP. As the smallest ground truth CP in the test set comprises 59 pixels, the defined minimum size does not affect the CP detection. The results are

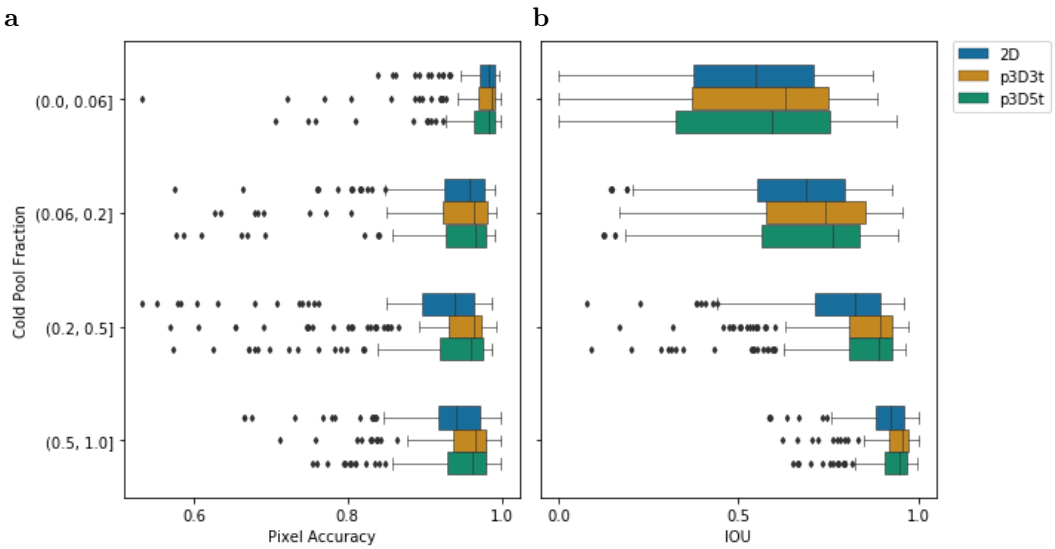

Figure 5: **Selected test performance metrics for varying cold pool fraction. a**, Distributions of pixel accuracy for each neural network, grouped into quartiles of cold pool fraction with ranges, as indicated along the vertical axis. Colored bars represent the interquartile range $IQR = Q3 - Q1$ of the three tested models, with the first quartile $Q1$ and the third quartile $Q3$, along with the corresponding median (vertical dash). Whiskers range from $Q1 - 1.5 \times IQR$ (minimum) to $Q3 + 1.5 \times IQR$ (maximum). Markers indicate outliers w.r.t. this range. **b**, Analogous to (a) but for the intersection over union (IOU) score. Note that for both metrics a value of unity reflects perfect accuracy, whereas zero denotes complete lack of accuracy.

Model	PA [%]	IOU	Precision	Recall
2D	93.8	0.71	0.84	0.83
p3D3t	94.8	0.75	0.83	0.88
p3D5t	94.5	0.74	0.84	0.87

Table 3: **Test performance of different models on patches with at least one pixel of both classes ("CP" / "no CP") in the ground truth.** Presented are mean performances for pixel accuracy (PA), intersection over union (IOU) score, Precision and Recall.

quite clear: Larger CPs are detected at quite high fidelity ($\gtrsim 90\%$), whereas the fidelity for the smallest area class is lower (Fig. 6). Again, a clear improvement in detection cannot be achieved for either of the three models, even though a slight improvement is seen for pseudo-3D models for the intermediate area classes.

In Tab. 6 we show the number of detected CPs for each simulation along with spuriously predicted CPs. Whereas the total number of detected CPs is slightly lower for the 2D network (483) than for p3D3t (495) and p3D5t (487), the p3D5t network features the highest number of spurious CPs (252), substantially more than those of 2D (199) and p3D3t (197). However, as the mean validation losses of the three p3D5t training runs are lowest in comparison to the other models (Tab. 2), this should not be a problem characteristic for p3D5t, but is most likely caused by an unfavorable epoch to stop the training run. Apart from lower detection rates for CPs from "diu2K", which are mainly at-

Model	PA [%]	σ(PA) [%]
2D	99.8	2.1
p3D3t	99.9	1.1
p3D5t	99.9	1.0

Table 4: **Test performance of different models on patches without any pixel of class "CP" in the ground truth.** Columns show the mean and standard deviation of pixel accuracy, PA for the various models.

Model	PA [%]	σ(PA) [%]
2D	92.0	9.5
p3D3t	94.1	9.2
p3D5t	85.9	15.5

Table 5: **Test performance of different models on patches with only pixel of class "CP" in the ground truth.** Analogous to Tab. 4.

tributed to a high proportion of CPs in the smallest area class, the performance of the networks seems to be relatively independent w.r.t. the simulation setup.

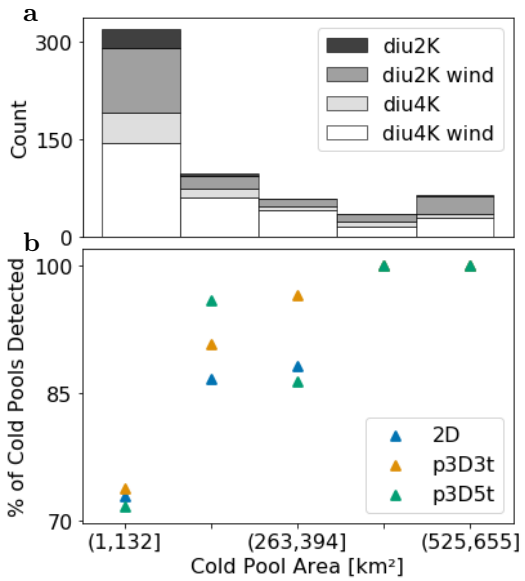

Figure 6: **Cold pool detection based on the test set. a**, Distribution of test set cold pools with respect to cold pool area; **b**, Percentage of successfully detected cold pools for varying cold pool area. A ground truth cold pool (CP) is considered detected if a predicted CP overlaps more than 50% of its area and if more than 50% of the area of the predicted CP falls inside the ground truth CP. Note the shared horizontal axis.

We now also characterize the spatial patterns detected by the neural network and compare them to the ground truth labeling. As the morphology of patterns is so diverse

Simulation	Total CPs	2D		p3D3t		p3D5t	
		Detected	False	Detected	False	Detected	False
diu2K	35	25	12	25	18	28	16
diu2K wind	180	158	52	154	62	156	66
diu4K	83	75	8	76	10	76	16
diu4K wind	292	225	127	240	107	227	154
Sum	590	483	199	495	197	487	252

Table 6: **Overview of detected test set cold pools for the different simulations.** A ground truth cold pool (CP) is considered detected if a predicted CP overlaps more than 50% of its area and if more than 50% of the area of the predicted CP falls inside the ground truth CP. All remaining predicted CPs are considered as False. Note that only patches with at least one CP in the ground truth were evaluated.

and quantification of spatial pattern overlap always requires to make choices as to the metrics used, we instead provide a qualitative discussion on typical cases. We visualize several predictions based on the test set and present 2D fields of rainfall intensity (r_{int}), cloud top temperature (T_B), the ground truth segmentation, as well as predictions of the three neural network models side by side (Fig. 7). The cases selected represent a range of circumstances: in some cases, cloud patterns are rather obvious and yield reasonable segmentation for all models (Fig. 7a). Where different aspects overlap temporally, such as cirrus from previous convection obscuring the present scene (Fig. 7b), all models may struggle with proper segmentation. Although cases with advection pose additional challenges, all models perform well for large CPs with large cloud-free areas, e.g., Fig. 7c. Yet, for cases in which the parent convection partly dissipated (Fig. 7d) or dissipates (Fig. 7e) pseudo-3D models give results which are physically more accurate w.r.t. the plausibility of the gust front. The same seems to be true for scenes with advected parent convection (Fig. 7f) — likely due to the fact that parts of the gust front are obscured when only using single patches, but revealed when taking a sequence of time steps into account. As a general outcome, all models perform reasonably well on the test cases described, yet, the distinction between 2D and pseudo-3D quality metrics is not as clear cut and should be assessed dependent on the scientific questions in focus.

4 Conclusion and Outlook

Cold pools likely play a key role in organizing the atmospheric convective cloud and precipitation field (Böing et al., 2012; Schlemmer & Hohenegger, 2016; Böing, 2016; Haerter et al., 2019; Haerter, 2019; Haerter et al., 2020; Nissen & Haerter, 2021; Muller et al., 2022). The present study demonstrates that cold pools can be detected via an artificial neural network by employing data readily available from geostationary satellite observations, namely cloud brightness temperature and precipitation. Altogether, using these two variables only, our networks were able to detect cold pools in data from cloud-resolving simulations with an overall mean accuracy between 93.8% (2D) and 94.8% (p3D3t) for patches with at least one pixel of both classes, $\geq$ 99.8% for patches without any pixel of class "CP", and between 85.9% (p3D5t) and 94.1% (p3D3t) for patches with only pixel of class "CP". Thus, we conclude that the method proposed should generally be suitable for the detection of cold pools from satellite data.

Robust detection of cold pool processes leading to the formation of MCS could be useful in better mechanistic understanding of organized convective systems and associ-

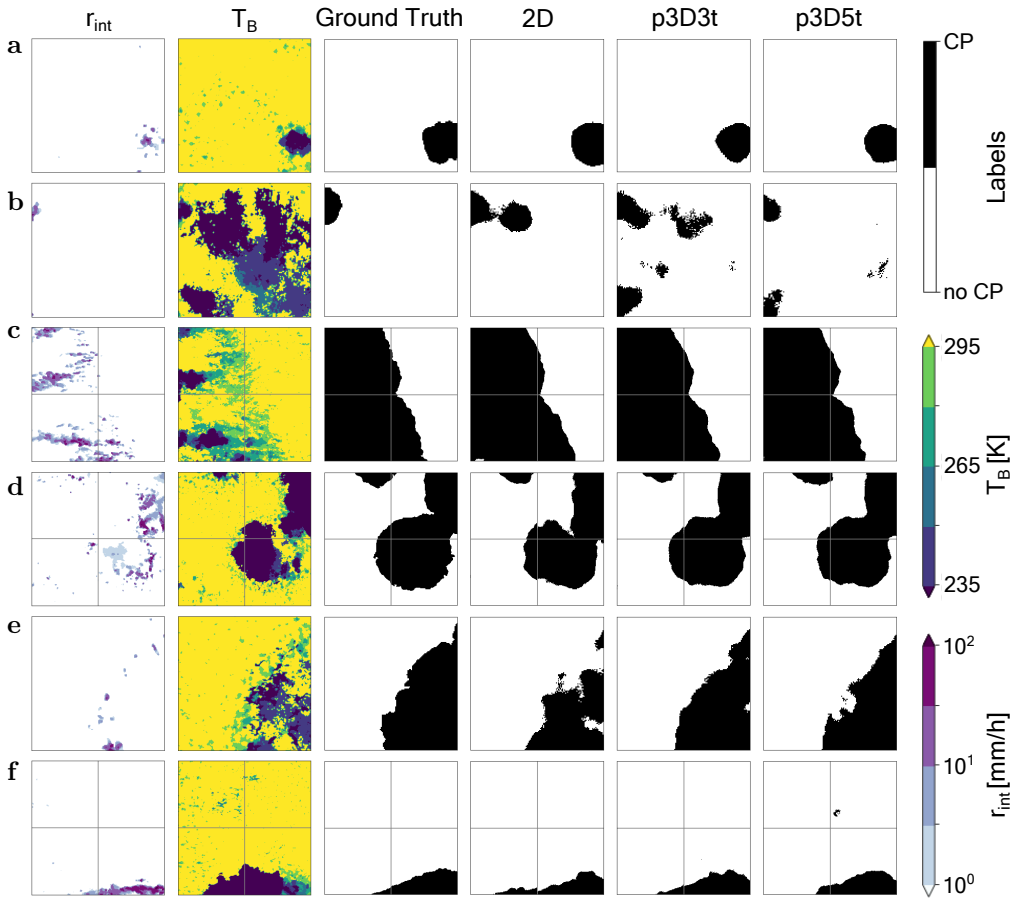

Figure 7: **Examples of cold pool predictions based on the test set.** Two-dimensional fields of surface rain intensity, r_{int}, and cloud top temperature T_B for various examples, along with ground truth segmentations based on CoolDeTA, as well as predictions of the 2D and pseudo-3D neural networks. **a**, Morning CP (time step 740) from "diu2K"; **b**, CP from "diu2K" which developed during the afternoon (time step 780) at the boundary of a recently dissipated convective system, represented by high-altitude cirrus remnants; **c**, parts of an eastward propagating gust front of a convective system from "diu2K wind" (time step 772) with large cloud free areas ($\gtrsim$ 300 km^2) and new emerging rain cells. The superimposed grid represents the individual $n_p \times n_p$ pixel patches, processed by the neural networks; **d**, afternoon scene (time step 772) from "diu4K" with parts of an early stage CP in the north of the upper left patch, and parts of a convective system which consists of CPs at different stages; **e**, Gust front of a convective system from "diu4K" (time step 780) with dissipating parent convection; **f**, Northern part of a CP from "diu4K wind" (time step 780) where westward advected parent convection masks parts of its CP gust front.

ated heavy precipitation events. We conducted several experiments to identify the most promising architecture for our network. The computationally most expensive architecture, using six blocks and 64 starting filters, performed best, as might be expected — given the physical insight that cold pools tend to grow to spatially and temporally correlated structures at the mesoscale, $\mathcal{O}(100\,km)$, forming MCS.

Including several time steps within the input channels is a computationally inexpensive means of mimicking a three-dimensional input data set. Whereas already the two-dimensional input fields gave satisfactory results, we find that taking into account

three to five time steps does improve the performance further, comparable to the improvements found in Vu et al. (Vu et al., 2020) for some of their data sets.

The comparison between weak and strong diurnal forcing is important, as it mimics cold pools both over ocean, where diurnal forcing is small, and over continents, where the diurnal range is large. Results show qualitatively different cloud organization, such as the formation of pronounced mesoscale convective systems over land but more scattered, smaller cold pools over the sea.

Assessing large-scale wind effects is important, as it compares the prominent model idealization of no wind shear (Manabe et al., 1965; Tompkins & Craig, 1998; Bretherton et al., 2005) with the more realistic sheared case. Our overall finding is that the detection works well for all these cases.

Looking ahead, the obvious next step is to apply the method to actual satellite data. Likely, several new challenges will need to be addressed, such as lower spatial and temporal resolution of the data available. The lower resolution may require us to focus on CPs that have already evolved into larger-scale structures. Yet, replacing the cloud brightness temperature input by the multiple individual satellite channels which are available, the neural network performance may benefit from patterns hidden so far. The combination of these satellite channels with a precipitation product based on calibrated infrared can avoid inconsistencies between inputs w.r.t. their spatial and temporal resolution. Ultimately, being able to extract self-organization effects from observational data will allow us to help improve cloud-resolving models that still struggle at capturing organizational effects at high fidelity. For this purpose, the network training should additionally focus on minimizing the number of spuriously predicted cold pools, e.g., by adding more examples of clouds to the training set which do not produce any cold pool. One way forward could be to further develop cold pool interaction parameterizations in coarser scale models.

Apart from applications related to observational data, our method could be adjusted for the detection of cold pools in simulation data for research and weather forecasting. In simulations we are not restricted to satellite-observable input variables. Provided as input to the network, variables such as temperature and moisture above the surface will likely enhance the accuracy of the network predictions further.

Acknowledgments
The authors gratefully acknowledge funding by a grant from the VILLUM Foundation (grant number: 13168) and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant number: 771859) and the Novo Nordisk Foundation Interdisciplinary Synergy Program (grant no. NNF19OC0057374). This work used resources of the Deutsches Klimarechenzentrum (DKRZ), granted by its Scientific Steering Committee (WLA) under project ID bb1166.

References

Benjamin, T. B. (1968). Gravity currents and related phenomena. *Journal of Fluid Mechanics*, *31*(2), 209–248.

Böing, S. J. (2016). An object-based model for convective cold pool dynamics. *Mathematics of Climate and Weather Forecasting*, *2*(1).

Böing, S. J., Jonker, H. J., Siebesma, A. P., & Grabowski, W. W. (2012). Influence of the subcloud layer on the development of a deep convective ensemble. *Journal of the Atmospheric Sciences*, *69*(9), 2682–2698. doi: https://doi.org/10.1175/JAS-D-11-0317.1

Bretherton, C. S., Blossey, P. N., & Khairoutdinov, M. (2005). An energy-balance analysis of deep convective self-aggregation above uniform SST. *Journal of the Atmospheric Sciences*, *62*(12), 4273–4292.

Drager, A. J., Grant, L. D., & van den Heever, S. C. (2020). Cold pool responses to changes in soil moisture. *Journal of Advances in Modeling Earth Systems*, *12*(8), e2019MS001922.

Drager, A. J., & van den Heever, S. C. (2017). Characterizing convective cold pools. *Journal of Advances in Modeling Earth Systems*, *9*(2), 1091–1115.

Droegemeier, K. K., & Wilhelmson, R. B. (1985). Three-dimensional numerical modeling of convection produced by interacting thunderstorm outflows. Part I: Control simulation and low-level moisture variations. *Journal of the Atmospheric Sciences*, *42*(22), 2381–2403.

Droegemeier, K. K., & Wilhelmson, R. B. (1987). Numerical simulation of thunderstorm outflow dynamics. part i: Outflow sensitivity experiments and turbulence dynamics. *Journal of the Atmospheric Sciences*, *44*(8), 1180–1210.

Feng, Z., Hagos, S., Rowe, A. K., Burleyson, C. D., Martini, M. N., & Szoeke, S. P. (2015). Mechanisms of convective cloud organization by cold pools over tropical warm ocean during the amie/dynamo field campaign. *J. Adv. Model. Earth Syst.*, *7*, 357-381. doi: 10.1002/2014ms000384

Fiévet, R., Meyer, B., & Haerter, J. O. (2022). On the sensitivity of convective cold pools to mesh resolution. *Earth and Space Science Open Archive*, 24. Retrieved from `https://doi.org/10.1002/essoar.10512297.1` doi: 10.1002/essoar.10512297.1

Fournier, M. B., & Haerter, J. O. (2019). Tracking the gust fronts of convective cold pools. *Journal of Geophysical Research: Atmospheres*, *124*(21), 11103–11117.

Fowler, H. J., Lenderink, G., Prein, A. F., Westra, S., Allan, R. P., Ban, N., ... others (2021). Anthropogenic intensification of short-duration rainfall extremes. *Nature Reviews Earth & Environment*, *2*(2), 107–122.

Fuglestvedt, H. F., & Haerter, J. O. (2020). Cold pools as conveyor belts of moisture. *Geophysical Research Letters*, *47*(12), e2020GL087319.

Gentine, P., Garelli, A., Park, S., Nie, J., Torri, G., & Kuang, Z. (2016). Role of surface heat fluxes underneath cold pools. *Geophys. Res. Lett.*, *43*, 874-883. doi: 10.1002/2015gl067262

Grandpeix, J.-Y., & Lafore, J.-P. (2010). A density current parameterization coupled with Emanuel's convection scheme. Part I: The models. *Journal of the Atmospheric Sciences*, *67*(4), 881–897.

Haerter, J. O. (2019). Convective self-aggregation as a cold pool-driven critical phenomenon. *Geophysical Research Letters*, *46*(7), 4017–4028. doi: https://doi.org/10.1029/2018GL081817

Haerter, J. O., Böing, S. J., Henneberg, O., & Nissen, S. B. (2019). Circling in on convective organization. *Geophysical Research Letters*, *46*(12), 7024–7034. doi: https://doi.org/10.1029/2019GL082092

Haerter, J. O., Meyer, B., & Nissen, S. B. (2020). Diurnal self-aggregation. *npj Climate and Atmospheric Science*, *3*. doi: 10.1038/s41612-020-00132-z

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

Henneberg, O., Meyer, B., & Haerter, J. O. (2020). Particle-based tracking of cold pool gust fronts. *J. Adv. Model. Earth Syst.*, *12*. doi: 10.1029/2019ms001910

Huffman, G. J., Bolvin, D. T., Braithwaite, D., Hsu, K., Joyce, R., Xie, P., & Yoo, S.-H. (2015). Nasa global precipitation measurement (gpm) integrated multi-satellite retrievals for gpm (imerg). *Algorithm Theoretical Basis Document (ATBD) Version*, *4*(26).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456).

Jadon, S. (2020). A survey of loss functions for semantic segmentation. In *2020 ieee conference on computational intelligence in bioinformatics and computational biology (cibcb)* (pp. 1–7).

Jensen, G. G., Fiévet, R., & Haerter, J. O. (2021). The diurnal path to persistent convective self-aggregation. *arXiv preprint arXiv:2104.01132*.

Khairoutdinov, M. F., & Randall, D. A. (2003). Cloud resolving modeling of the arm summer 1997 iop: Model formulation, results, uncertainties, and sensitivities. *Journal of Atmospheric Sciences*, *60*(4), 607–625.

Lundgren, T., Yao, J., & Mansour, N. (1992). Microburst modelling and scaling. *Journal of fluid mechanics*, *239*, 461–488.

Manabe, S., Smagorinsky, J., & Strickler, R. F. (1965). Simulated climatology of a general circulation model with a hydrologic cycle. *Mon. Wea. Rev*, *93*(12), 769–798.

Markowski, P., & Richardson, Y. (2011). *Mesoscale meteorology in midlatitudes* (Vol. 2). John Wiley & Sons.

Meyer, B., & Haerter, J. O. (2020, 11). Mechanical forcing of convection by cold pools: Collisions and energy scaling. *J. Adv. Model. Earth Syst.*, *12*(11), n/a–n/a. Retrieved from `https://doi.org/10.1029/2020MS002281` doi: 10.1029/2020MS002281

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3d vision (3dv)* (pp. 565–571).

Muller, C., Yang, D., Craig, G., Cronin, T., Fildier, B., Haerter, J. O., ... others (2022). Spontaneous aggregation of convective storms. *Annual Review of Fluid Mechanics*, *54*, 133–157.

Nissen, S. B., & Haerter, J. O. (2021). Circling in on convective self-aggregation. *Journal of Geophysical Research: Atmospheres*, *126*(20), e2021JD035331.

Redl, R., Fink, A. H., & Knippertz, P. (2015). An objective detection method for convective cold pool events and its application to northern africa. *Monthly Weather Review*, *143*, 5055-5072. doi: 10.1175/mwr-d-15-0223.1

Rio, C., Hourdin, F., Grandpeix, J.-Y., & Lafore, J.-P. (2009). Shifting the diurnal cycle of parameterized deep convection over land. *Geophysical Research Letters*, *36*(7).

Romps, D. M., & Jeevanjee, N. (2016). On the sizes and lifetimes of cold pools. *Quarterly Journal of the Royal Meteorological Society*, *142*(696), 1517–1527.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks

for biomedical image segmentation. In *Medical image computing and computer-assisted intervention (miccai)* (Vol. 9351, pp. 234–241). Springer. Retrieved from https://doi.org/10.1007/978-3-319-24574-4_28 doi: 10.1007/978-3-319-24574-4_28

Schlemmer, L., & Hohenegger, C. (2016). Modifications of the atmospheric moisture field as a result of cold-pool dynamics. *Quarterly Journal of the Royal Meteorological Society*, *142*, 30–42. doi: https://doi.org/10.1002/qj.2625

Sharifnezhadazizi, Z., Norouzi, H., Prakash, S., Beale, C., & Khanbilvardi, R. (2019). A global analysis of land surface temperature diurnal cycle using modis observations. *Journal of Applied Meteorology and Climatology*, *58*(6), 1279–1291.

Shi, J., Dang, J., Cui, M., Zuo, R., Shimizu, K., Tsunoda, A., & Suzuki, Y. (2021). Improvement of damage segmentation based on pixel-level data balance using vgg-unet. *Applied Sciences*, *11*(2), 518.

Simpson, J. (1980). Downdrafts as linkages in dynamic cumulus seeding effects. *Journal of Applied Meteorology*, *19*(4), 477–487.

Tan, J., Jakob, C., Rossow, W. B., & Tselioudis, G. (2015). Increases in tropical rainfall driven by changes in frequency of organized deep convection. *Nature*, *519*(7544), 451–454. doi: https://doi.org/10.1038/nature14339

Tompkins, A. M. (2001). Organization of tropical convection in low vertical wind shears: The role of cold pools. *Journal of the Atmospheric Sciences*, *58*, 1650-1672. doi: 10.1175/1520-0469(2001)058<1650:ootcil>2.0.co;2

Tompkins, A. M., & Craig, G. C. (1998). Radiative–convective equilibrium in a three-dimensional cloud-ensemble model. *Quarterly Journal of the Royal Meteorological Society*, *124*(550), 2073–2097. doi: https://doi.org/10.1002/qj.49712455013

Torri, G., & Kuang, Z. (2016). Rain evaporation and moist patches in tropical boundary layers. *Geophysical Research Letters*, *43*(18), 9895–9902.

Torri, G., & Kuang, Z. (2019). On cold pool collisions in tropical boundary layers. *Geophys. Res. Lett.*, *46*, 399-407. doi: 10.1029/2018gl080501

Torri, G., Kuang, Z., & Tian, Y. (2015). Mechanisms for convection triggering by cold pools. *Geophysical Research Letters*, *42*(6), 1943–1950.

Vu, M. H., Grimbergen, G., Nyholm, T., & Löfstedt, T. (2020). Evaluation of multislice inputs to convolutional neural networks for medical image segmentation. *Med. Phys.*, *47*, 6216-6231. doi: 10.1002/mp.14391

Willemink, M. J., Koszek, W. A., Hardell, C., Wu, J., Fleischmann, D., Harvey, H., ... Lungren, M. P. (2020). Preparing medical imaging data for machine learning. *Radiology*, *295*(1), 4–15.

Yamaguchi, T., Randall, D. A., & Khairoutdinov, M. F. (2011). Cloud modeling tests of the ultimate–macho scalar advection scheme. *Monthly Weather Review*, *139*(10), 3248–3264.

Zuidema, P., Torri, G., Muller, C., & Chandra, A. (2017). A survey of precipitation-induced atmospheric cold pools over oceans and their interactions with the larger-scale environment. *Surveys in Geophysics*, 1–23.