人工智慧於短延時降雨預測之應用

AI in Nowcasting

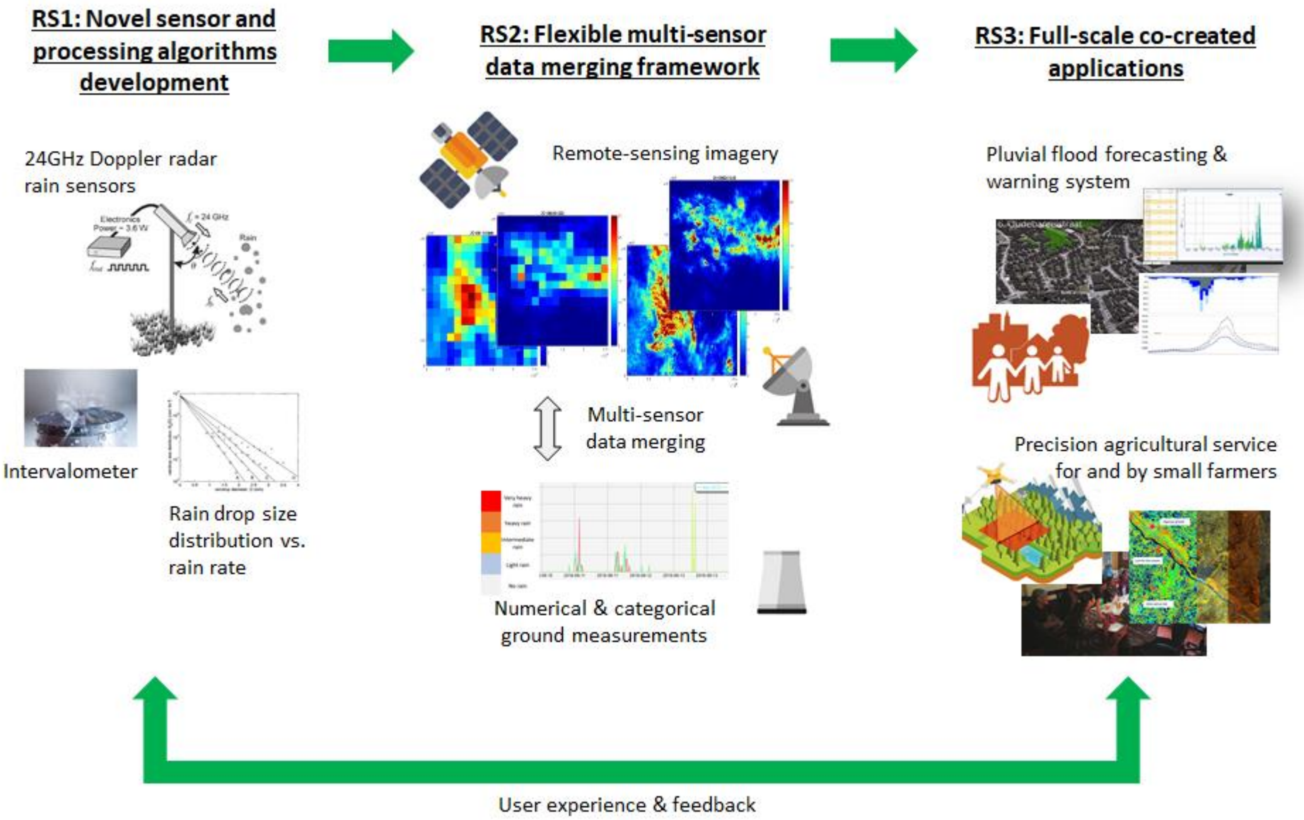
- Some insights gained from reproducing DGMR

- Li-Pen Wang (lpwang@ntu.edu.tw)
- Yun-Ting Heh (hehyunting@caece.net)

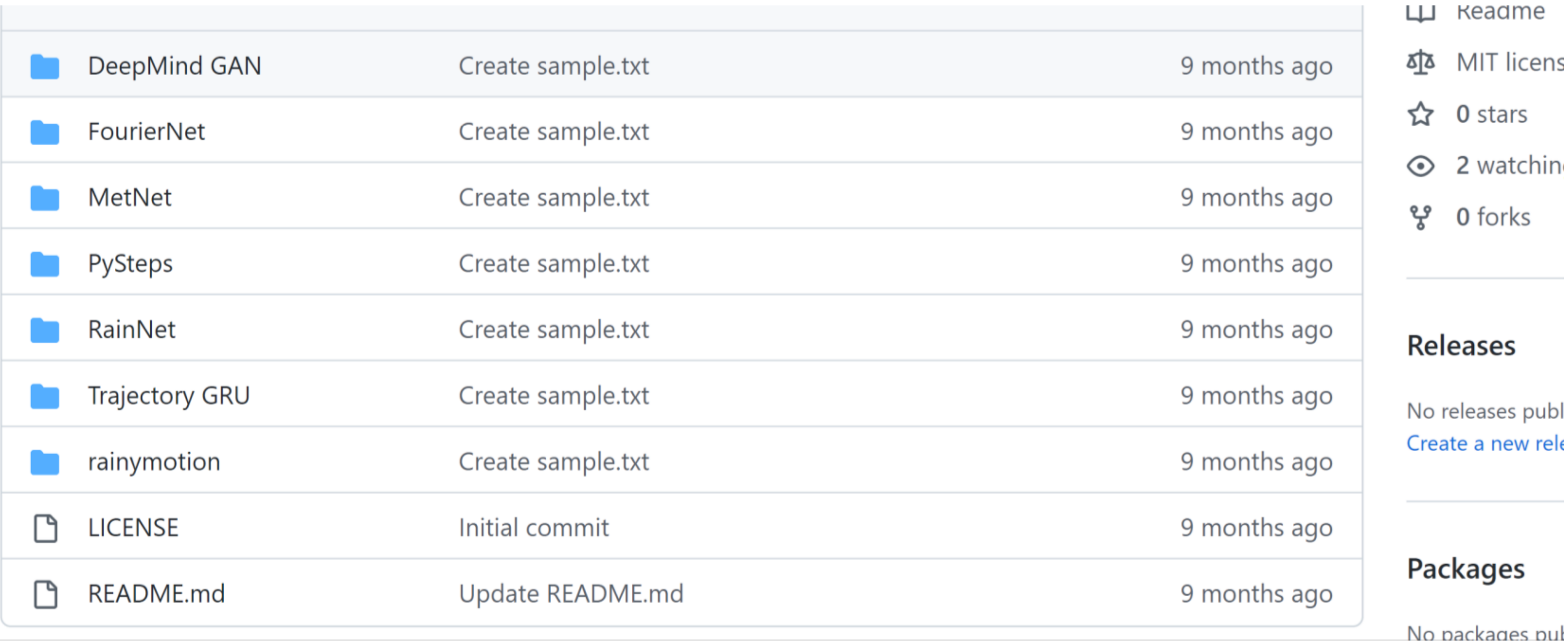
Computational Hydrometeorological Lab
Department of Civil Engineering, National Taiwan University

SWCB seminar, 4th October 2022

Novel sensing, algorithms & interfacing



A joint effort from the entire Computational Hydrometeorological Lab



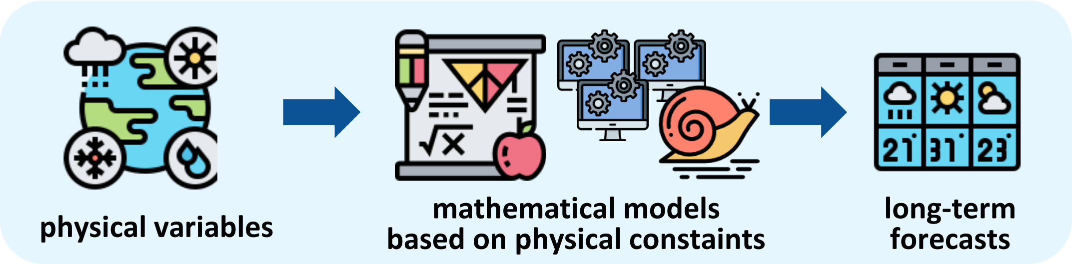
It started from an internal team project 2 years ago…

Some challenges in modelling spatial-temporal rainfall process

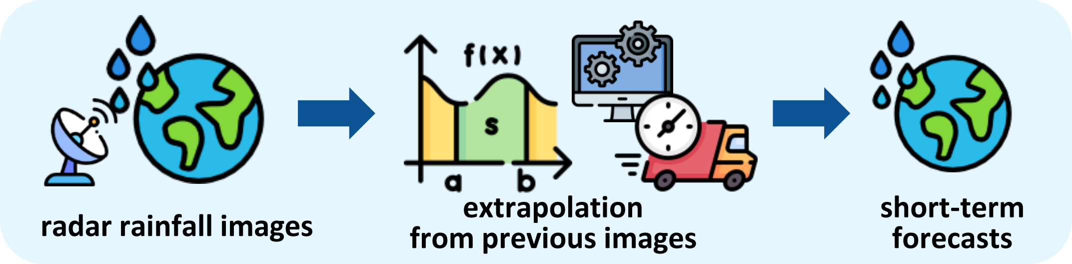
- **Variations in rainfall = Advection + Evolution (in time)**
- **Spatial and temporal features of rainfall are not independent from each other**
- **Scale matters!**

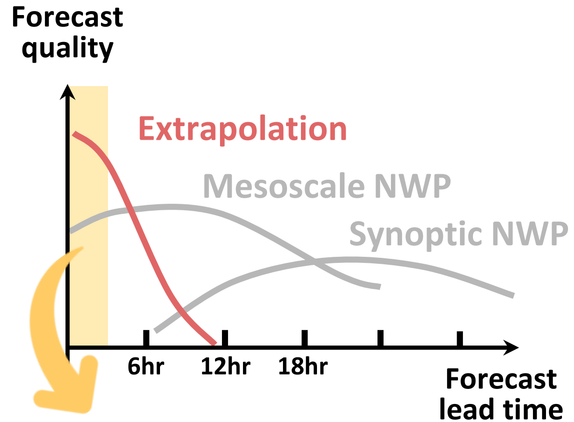
Radar-based nowcasting is effective and more affordable

❑ **Numerical weather prediction system (NWP)**



❑ **Radar-based nowcasting**





Target lead time of nowcasting

2~3 hours

For nowcasting purpose, radar-based methods are more effective than NWP.

Field-based nowcasting

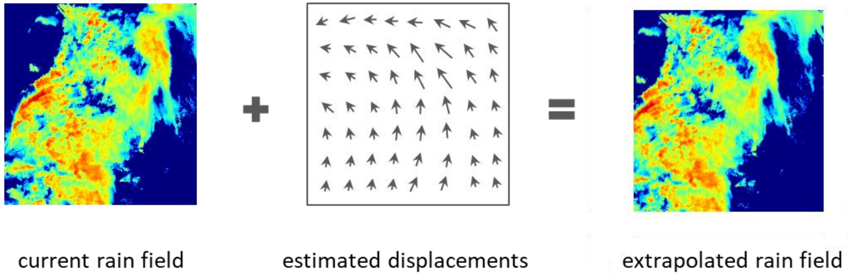
❑ Advection nowcasting

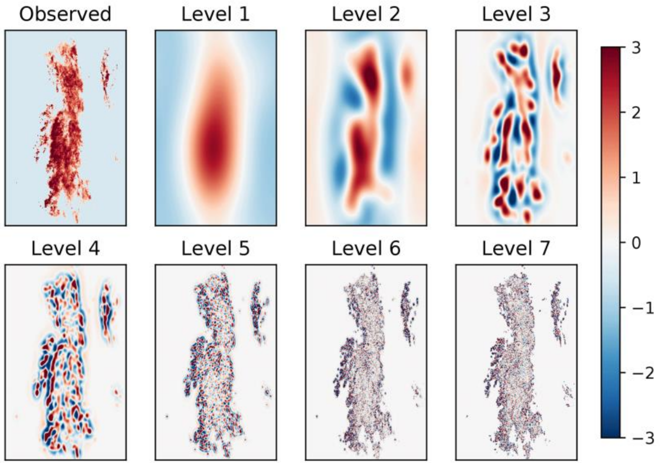
- Extrapolates current radar rainfall fields into future frames using estimated displacements

❑ STEPS

- State-of-the-art nowcasting model
- **Optical flow** is employed for displacement estimation
- Probabilistic nowcasting model
- The spatial-temporal scaling relationship is explicitly modelled.

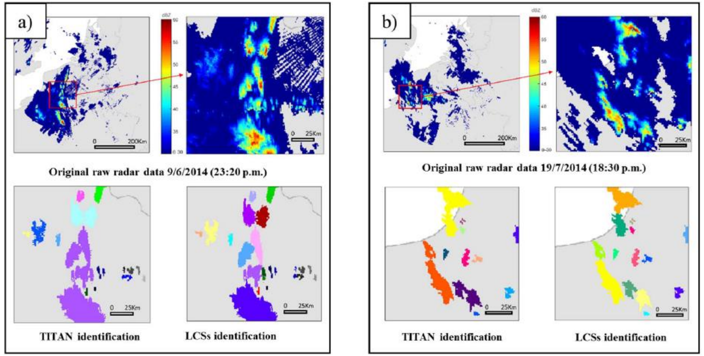
https://pysteps.github.io/

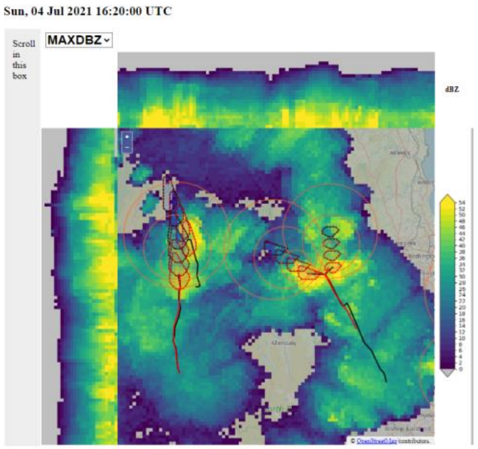




Object-based nowcasting

- ❑ **Object-based nowcasting**
  - Extrapolates the movement of identified rainfall cells
- ❑ **TITAN**
  - Most widely-used basis model
  - Storm identification
  - Temporal association of rainfall objects between successive time steps
  - Widely used for thunderstorm nowcasting



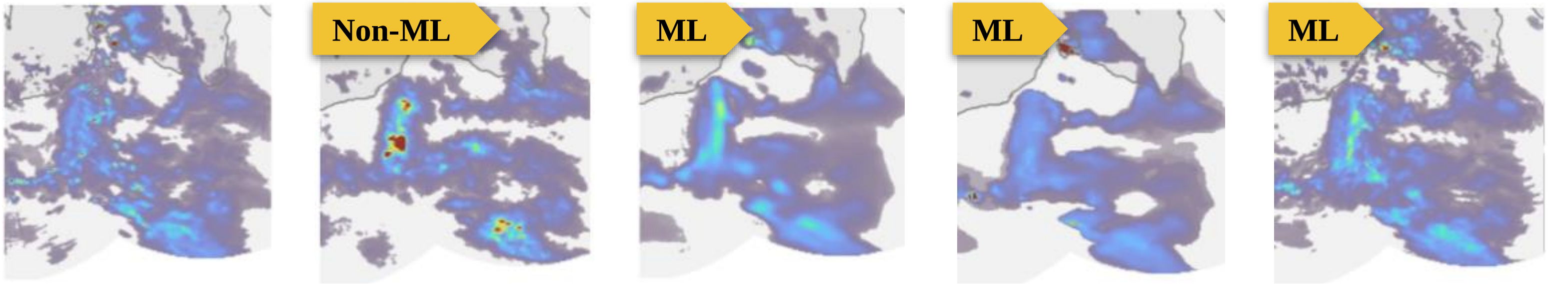


It is NOT new to apply AI to rainfall nowcasting,

but paradigm shifts as new technologies emerge.

DGMR: Deep Generative Models of Radar proposed by DeepMind

In 2021, DeepMind proposed a deep-learning model, ***Deep Generative Models of Radar***, that achieved a great success in the field of **short-term rainfall nowcasting** (Ravuri et al., 2021).



Ground truth **STEPS** **UNet** **MetNet** **DGMR**

Figure. Case study of performance on a challenging precipitation event starting on = 24 June 2019 at 16:15 UK, showing convective cells over eastern Scotland (Ravuri et al., 2021).

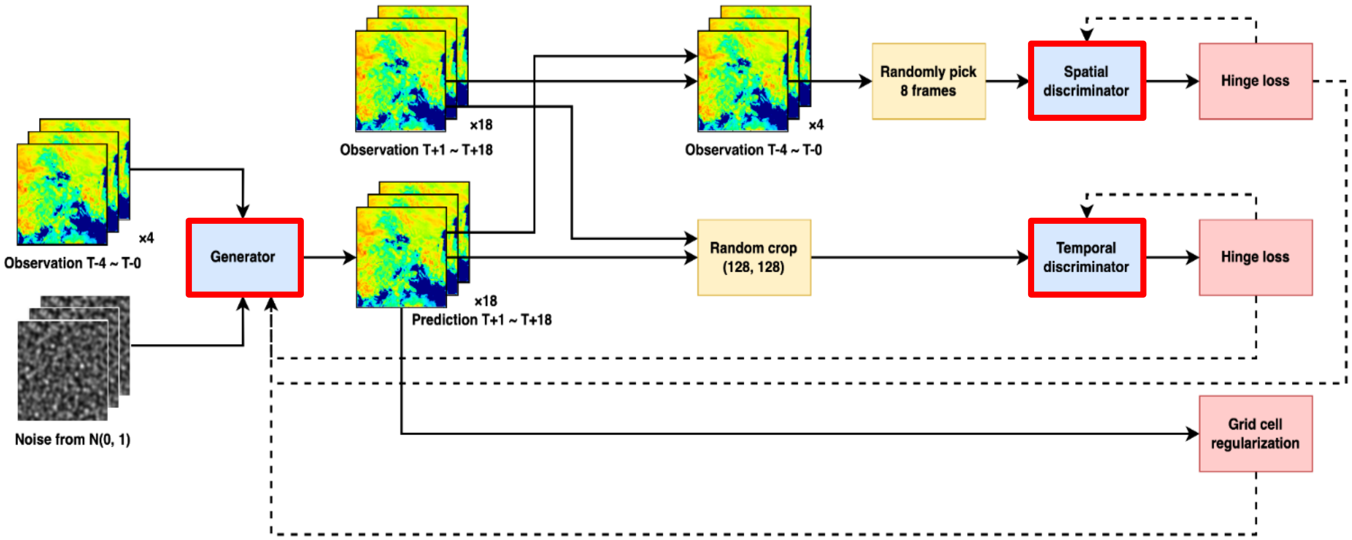
Check out DGMR's article!

DGMR: A rainfall nowcasting model trained with the GAN technique

DGMR includes three main models, which are trained simultaneously via an adversarial process.

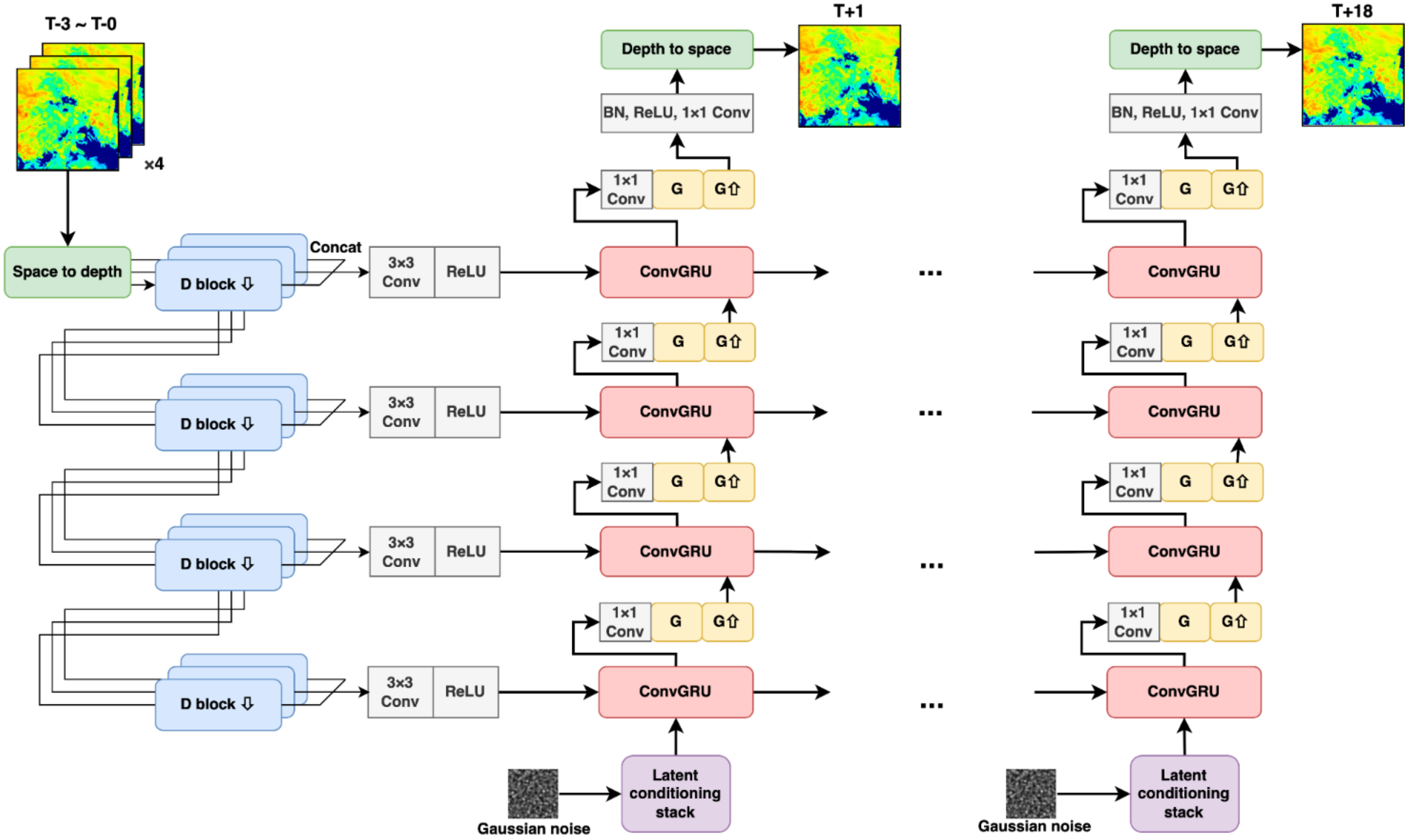
These models are:

- **A Generator** that produces rainfall nowcasts.
- **Two Discriminators** that discriminate if the generated nowcasts are ‘similar enough’ to the ground truth in terms of their spatial and temporal features, respectively.



Check out DGMR’s article!

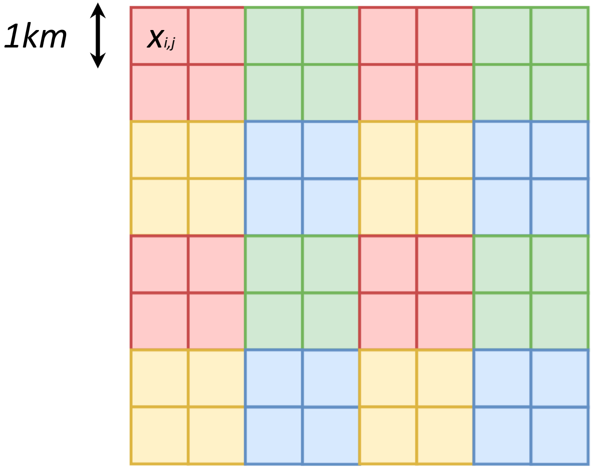
Overview of the DGMR generator



Tutorial Outline

- **What is Convolution?**
  - How does it work on a 2-dimensional images?
- **What is Space to Depth?**
  - How does it work?
  - Space to Depth + Convolution
  - Hierachical feature extractor
- **What is GRU (Gated Recurrent Units)?**
  - How does GRU work?
  - The role of ConvGRU in DGMR
- **What is GAN?**
  - GAN framework in a nutshell
  - The benefit of using GAN

Convolution: How does it work on a 2D image?



original image

Convolution mask

$w_{0,0}$	$w_{0,1}$	$w_{0,2}$
$w_{1,0}$	$w_{1,1}$	$w_{1,2}$
$w_{2,0}$	$w_{2,1}$	$w_{2,2}$

As mentioned before…

$F_0(x) = \Sigma\, w_n\, x^n$

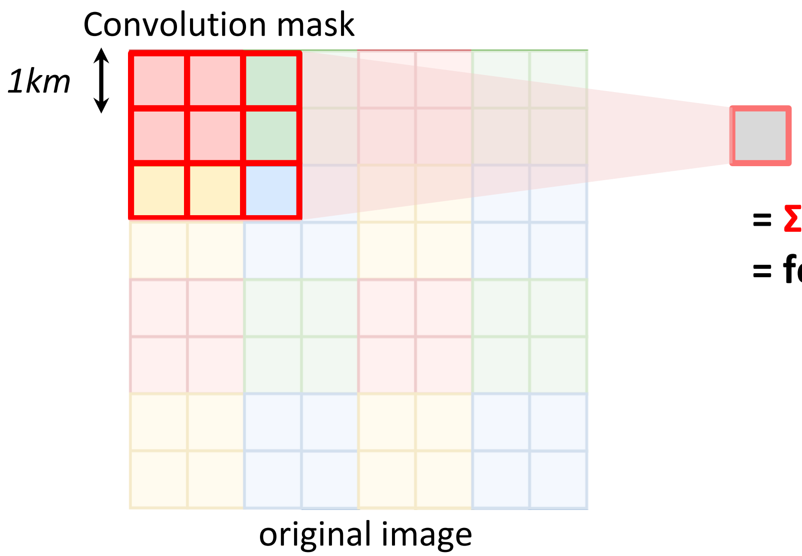
$F_1(x) = \Sigma\, w_n\, x^n$

$F_2(x) = \Sigma\, w_n\, x^n$

⋮

$F_j(x) = \Sigma\, w_n\, x^n$

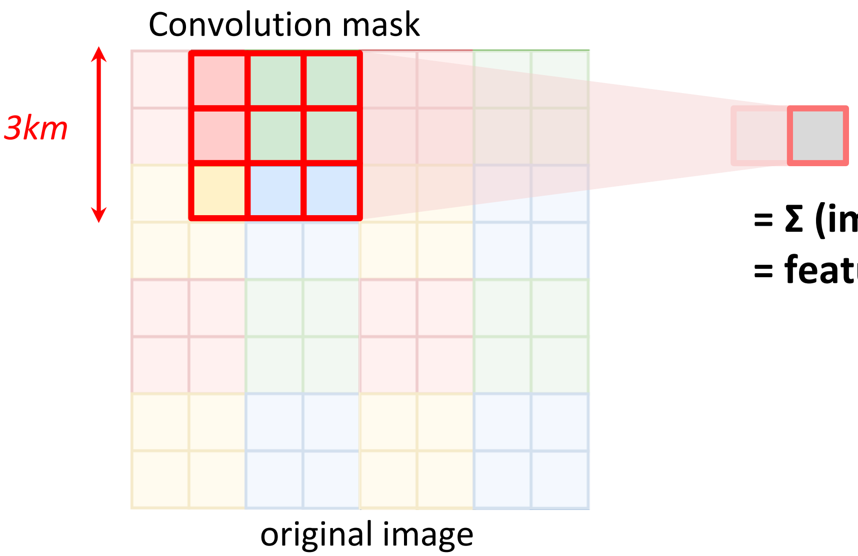
Convolution: How does it work on a 2D image?



= **Σ (image pixel × mask pixel)**

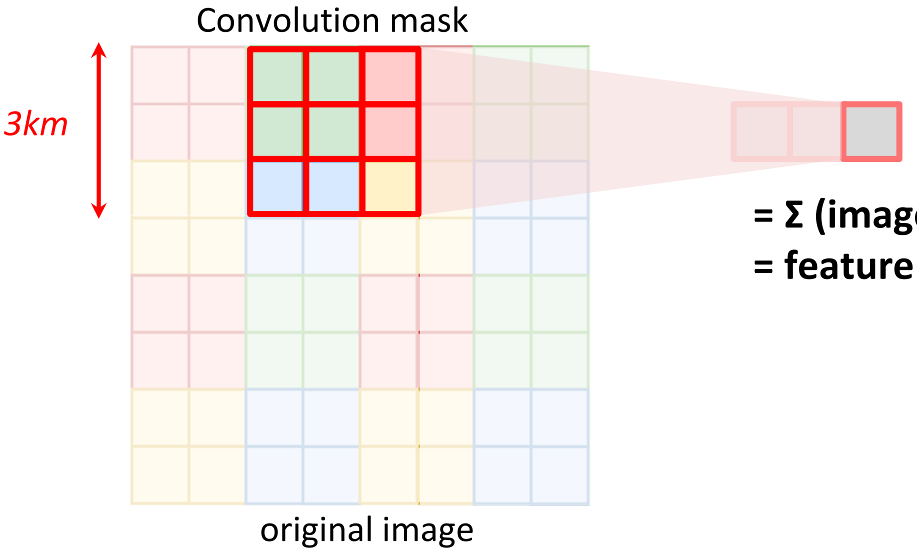
= **feature**

Convolution: How does it work on a 2D image?



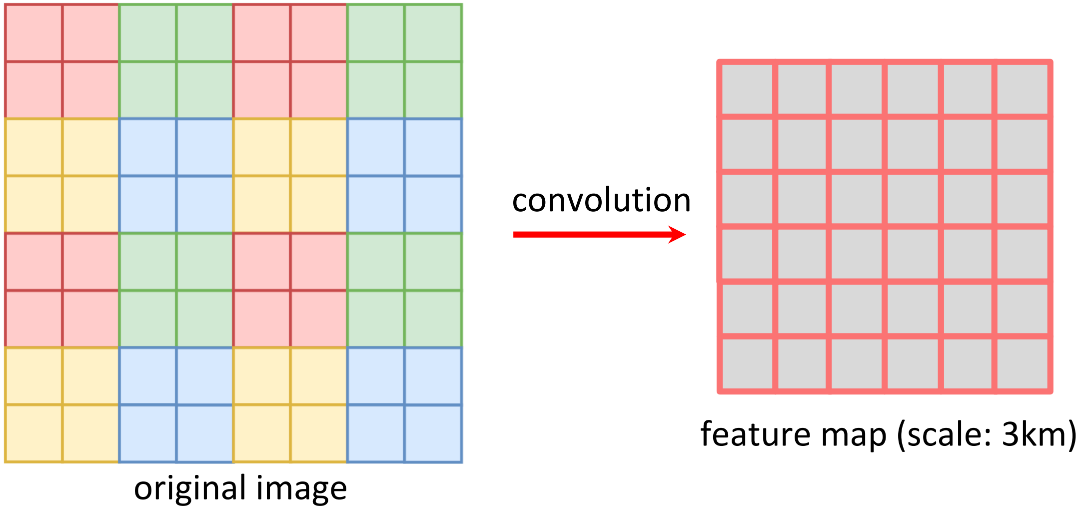
= Σ (image pixel × mask pixel)
= feature

Convolution: How does it work on a 2D image?

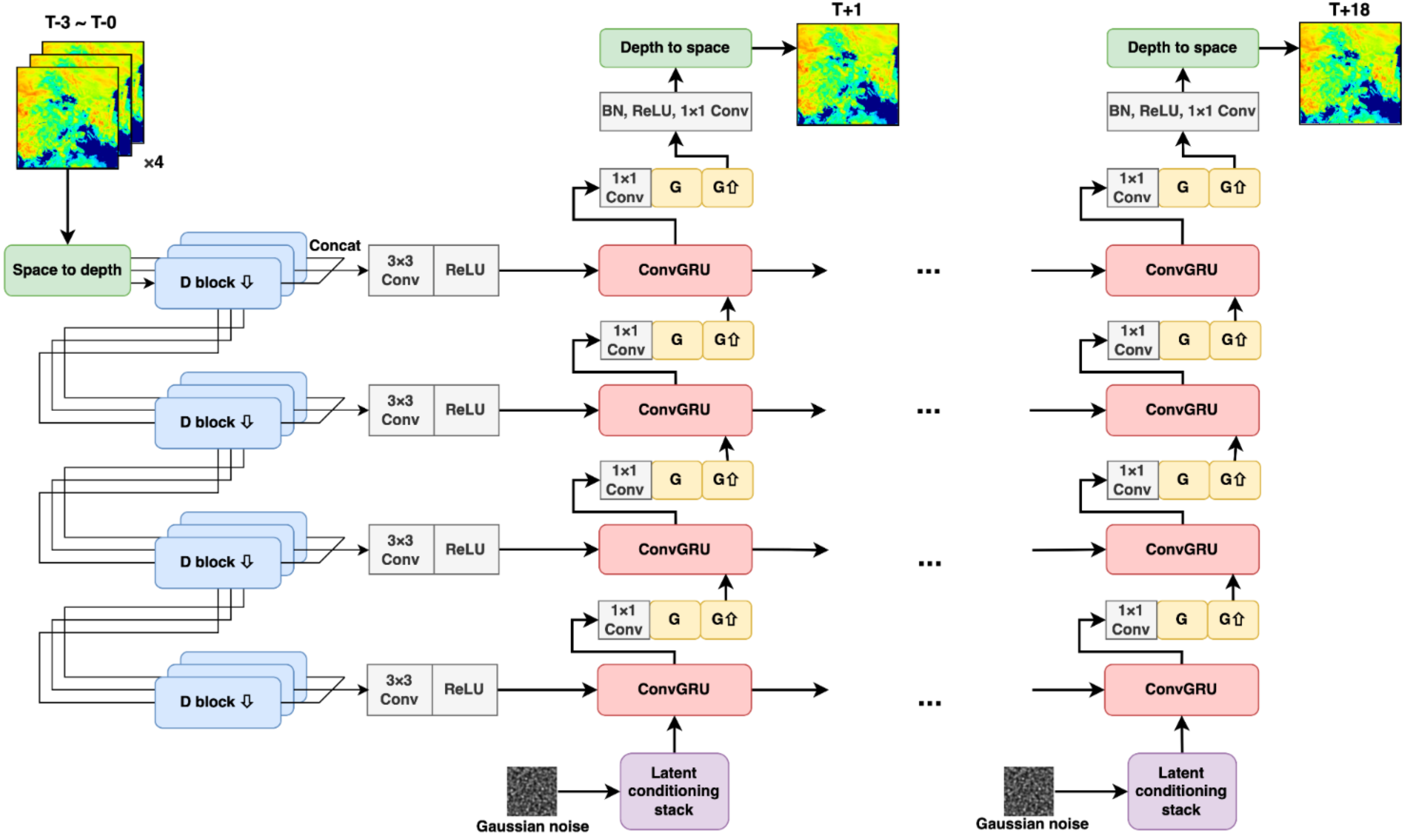


= Σ (image pixel × mask pixel)
= feature

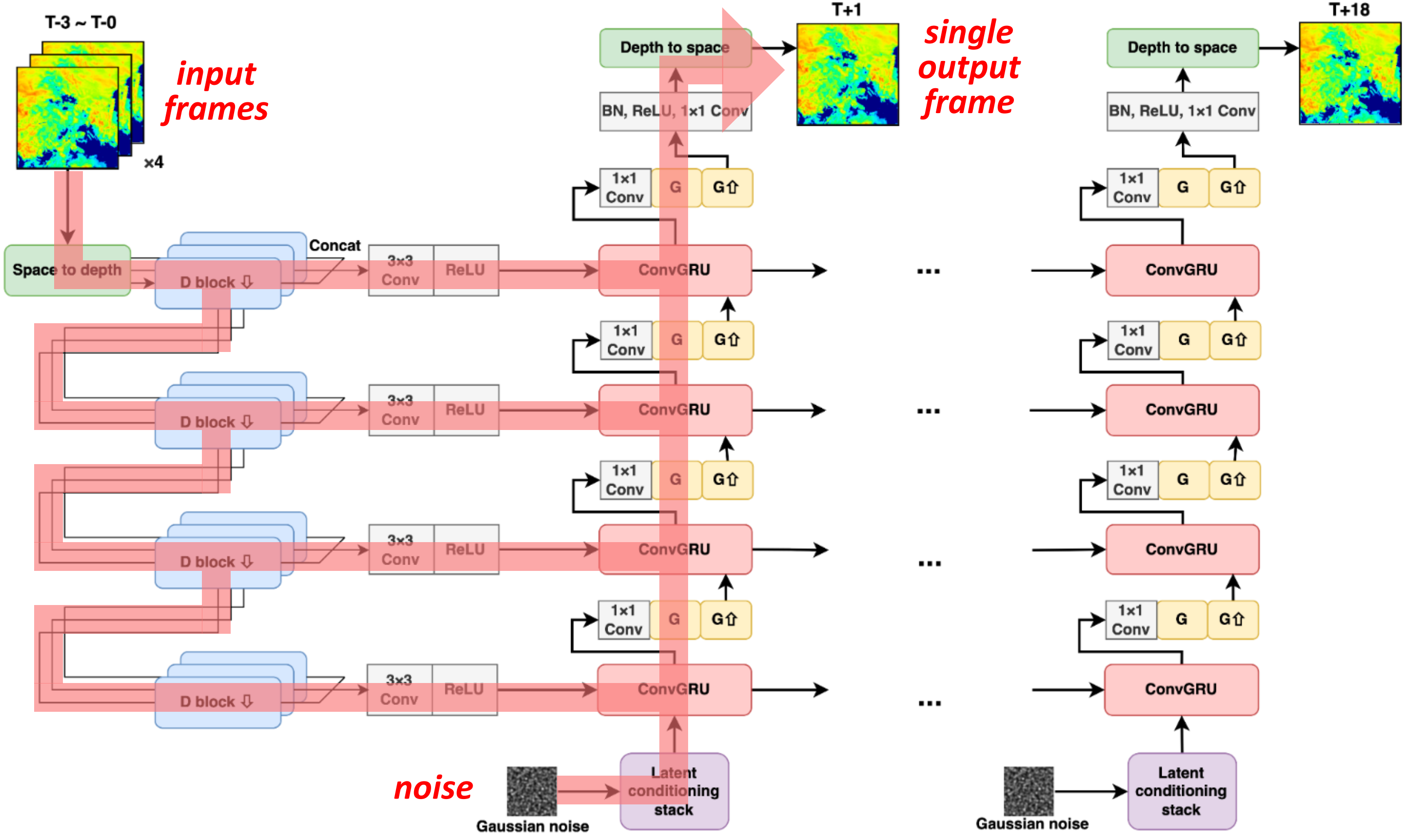
Convolution: How does it work on a 2D image?



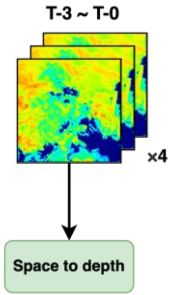
Deep Generative Models of Radar proposed by DeepMind (DGMR): An Overview



DGMR: An Overview

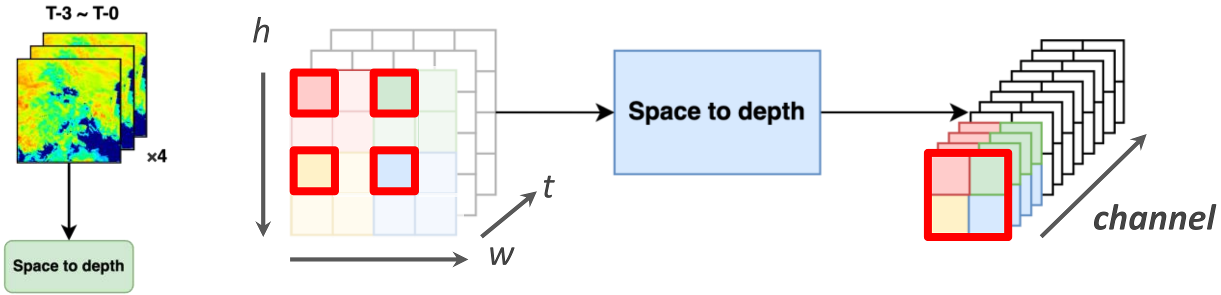


Space to Depth (S2D): How does it work?



What does *Space to depth* do?

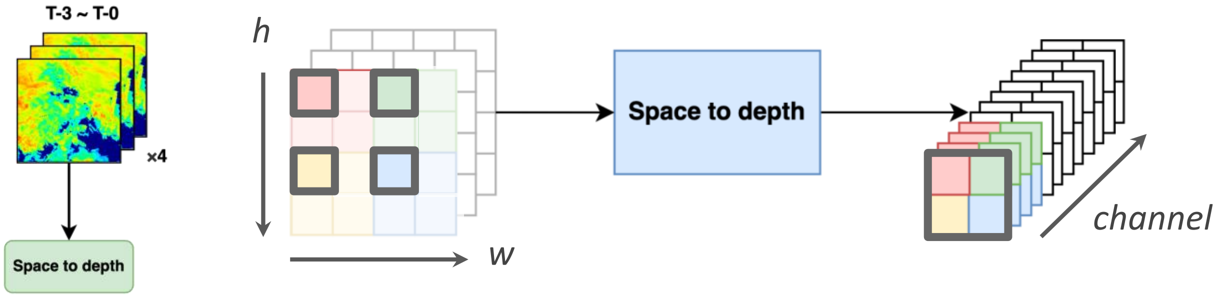
Space to Depth (S2D): How does it work?

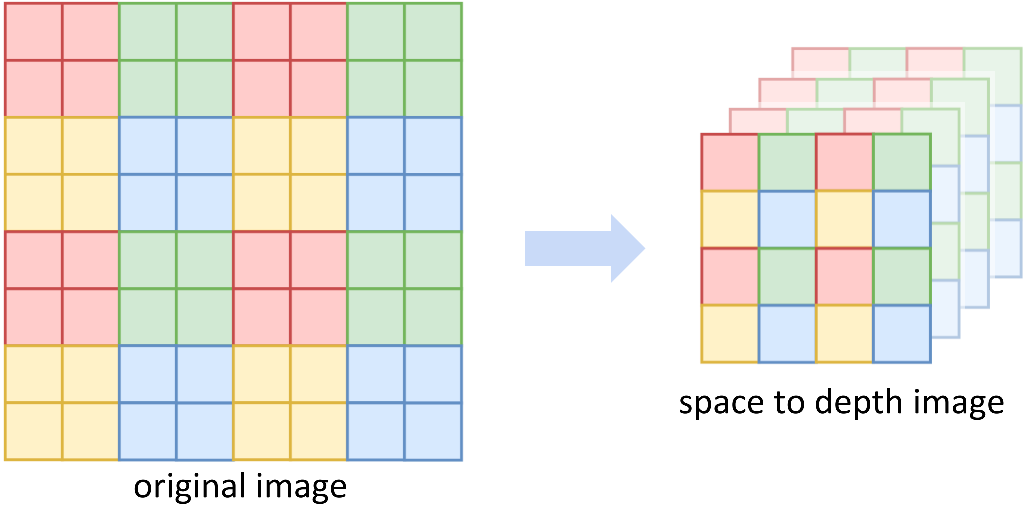


What does *Space to depth* do?

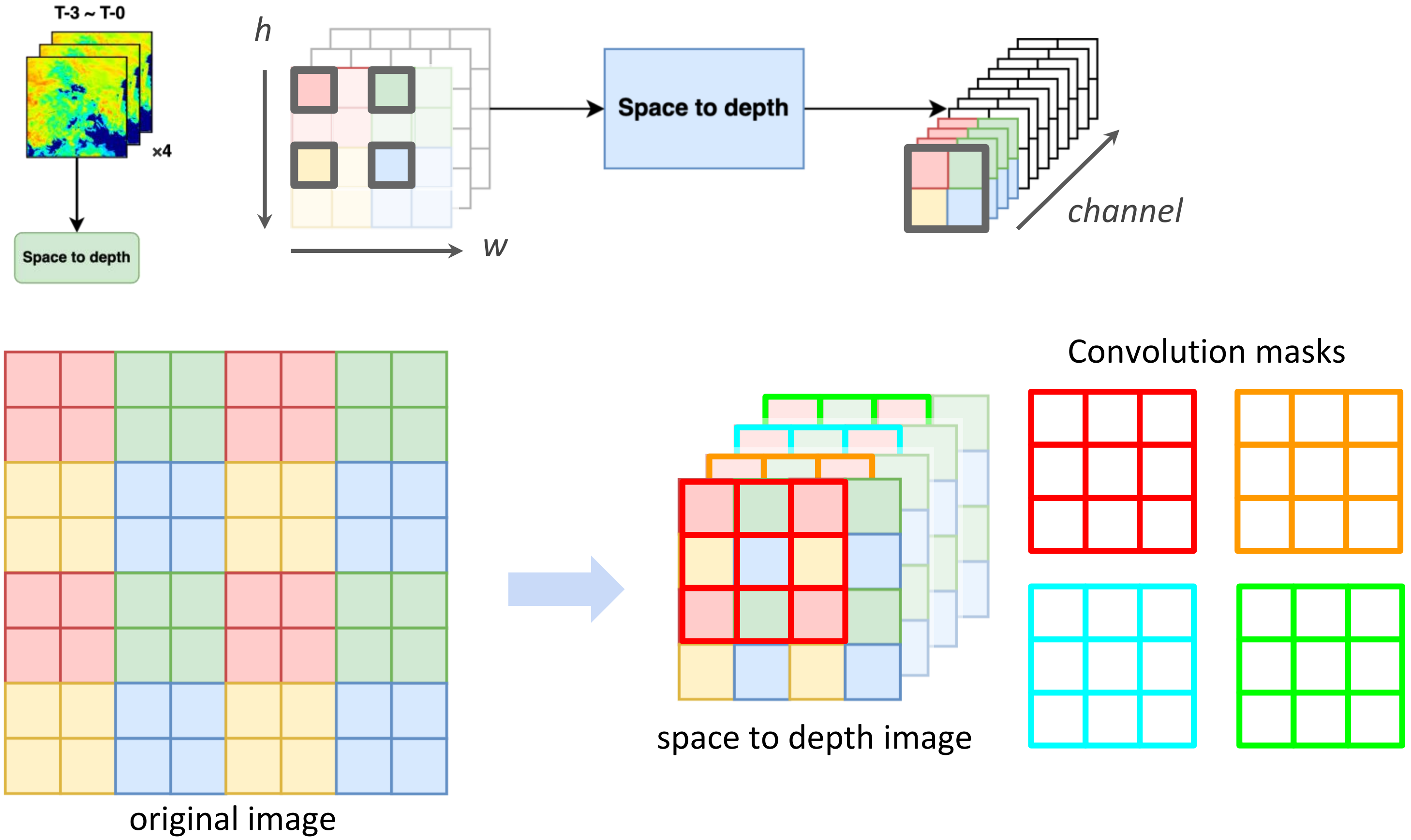
Relocate *height dimension* and *width dimension* data to *time dimension.*

Space to Depth + Convolution

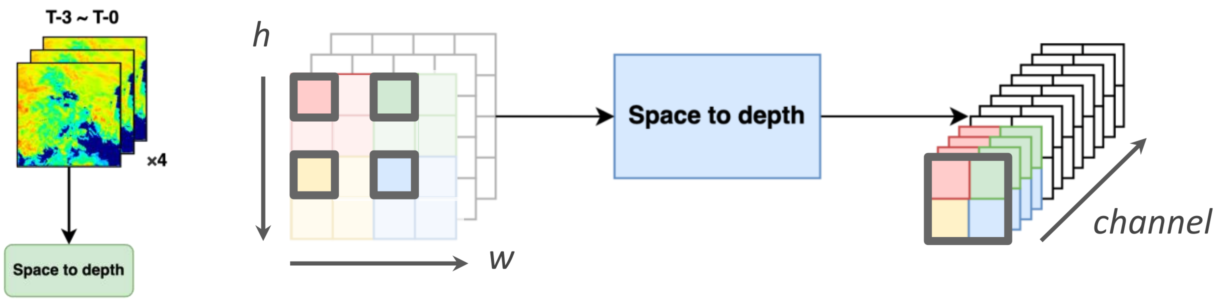


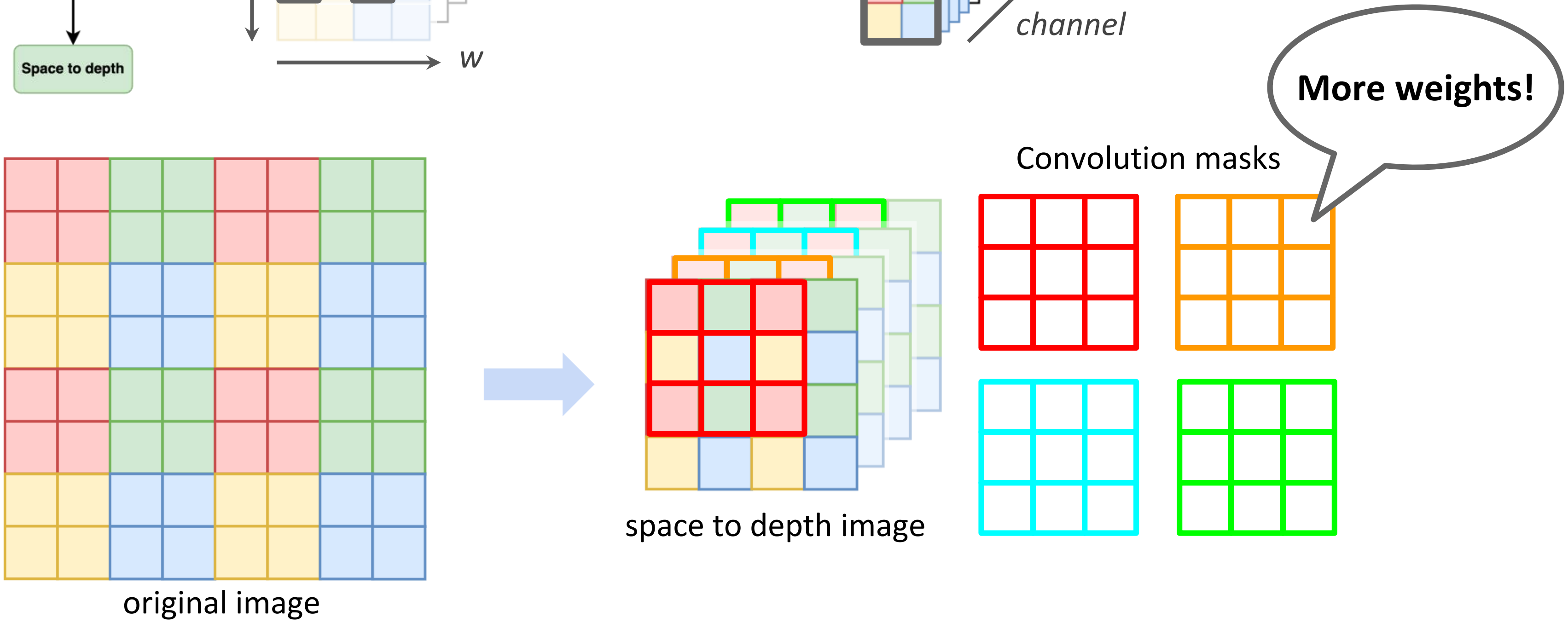


Space to Depth + Convolution: More learnable weights.

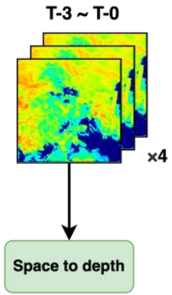


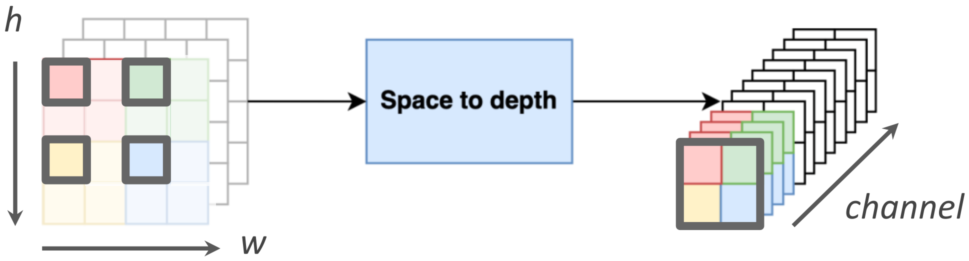
Space to Depth + Convolution: More learnable weights.

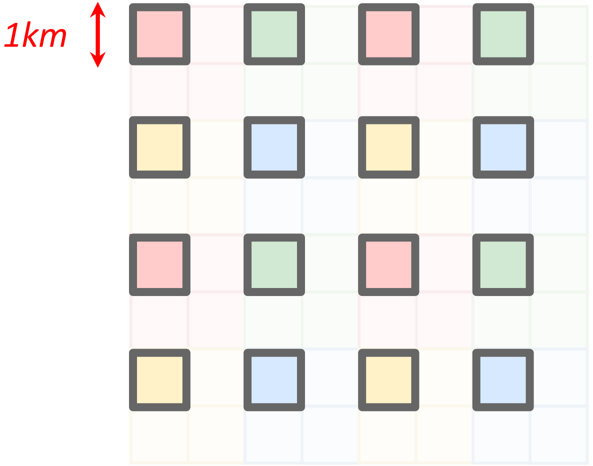




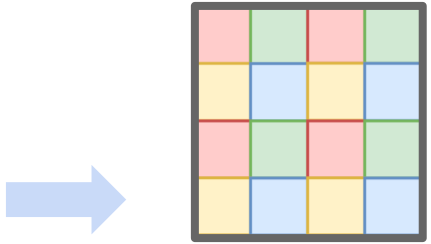
Space to Depth + Convolution







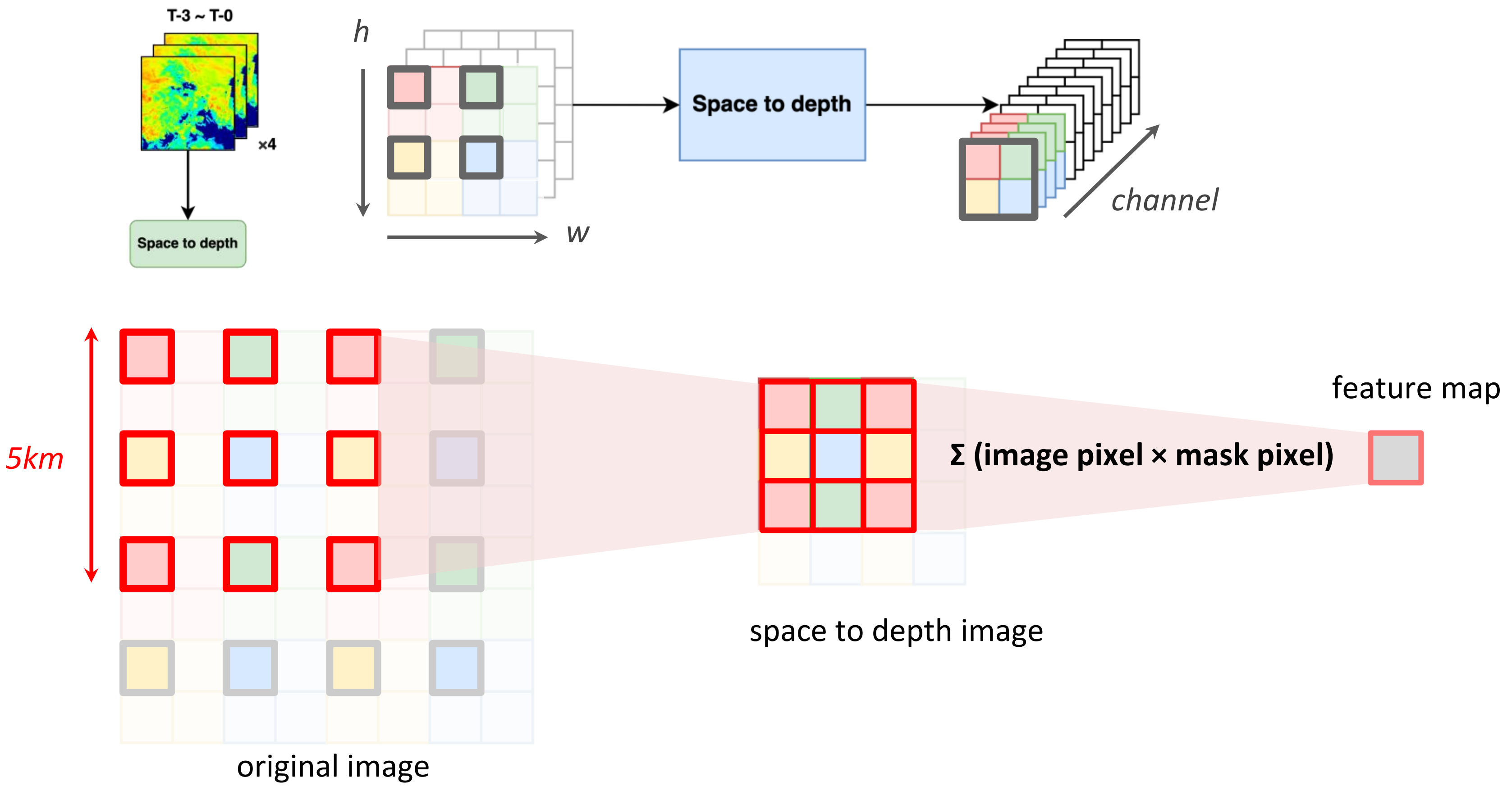
original image



space to depth image

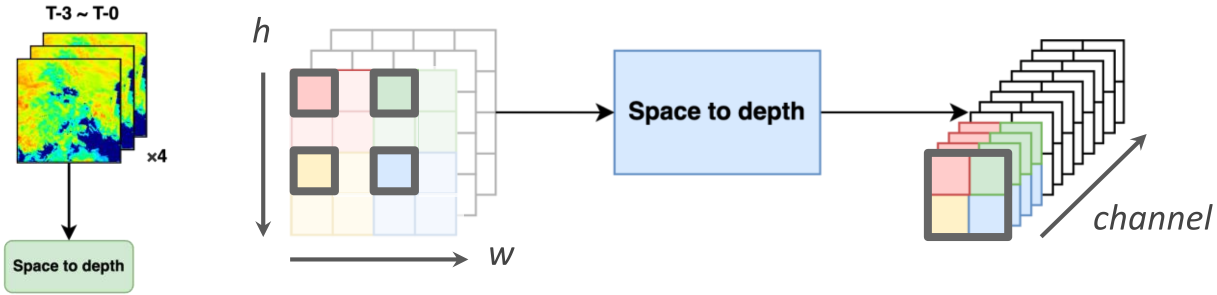
Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

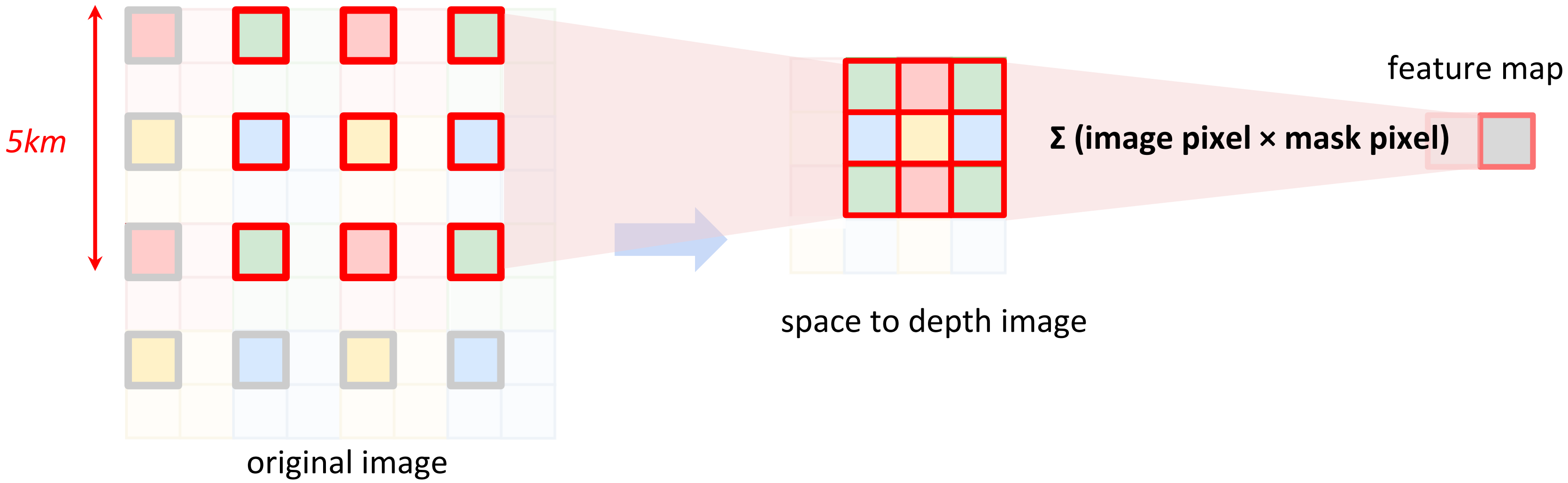
Space to Depth + Convolution: How features are extracted?



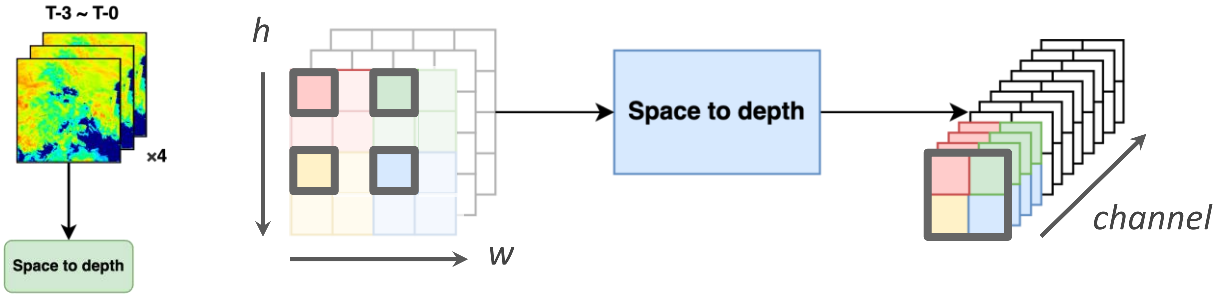
Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

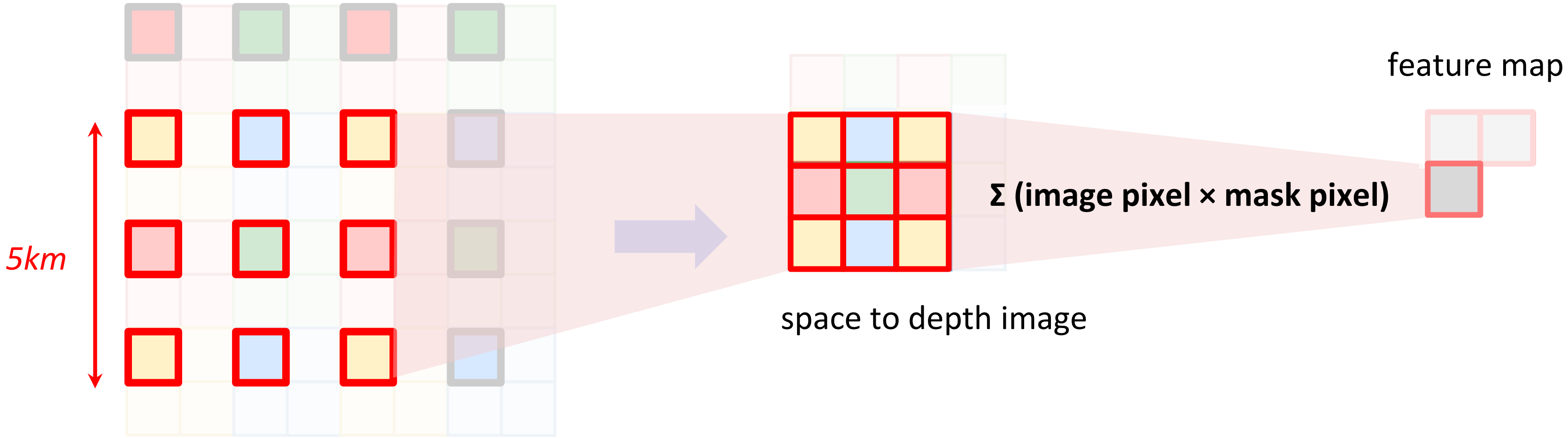
Space to Depth + Convolution: How features are extracted?





Space to Depth + Convolution: How features are extracted?

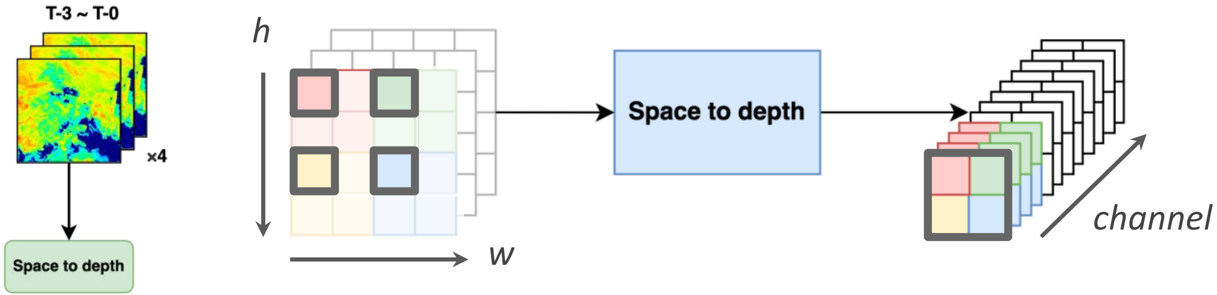


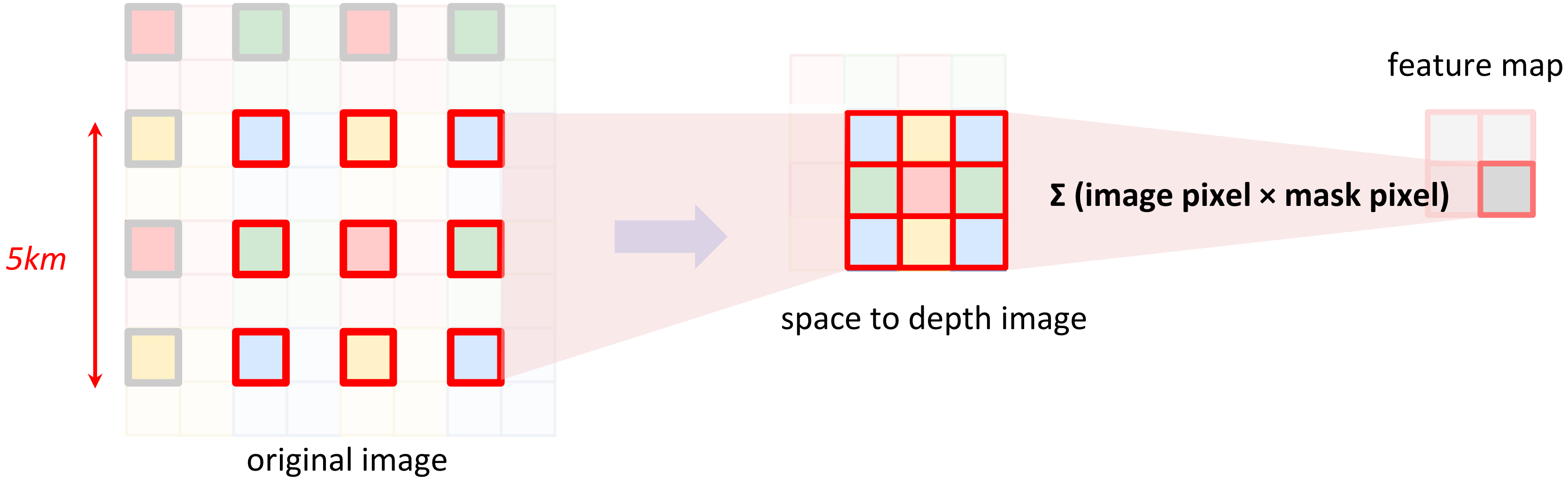


original image

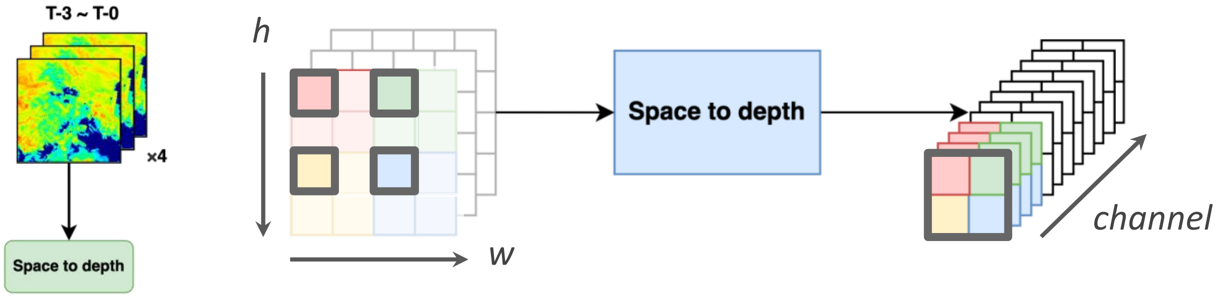
space to depth image

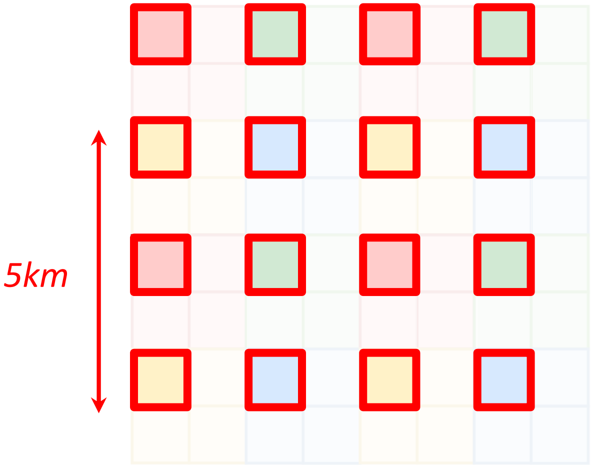
Space to Depth + Convolution: How features are extracted?





Space to Depth + Convolution: How features are extracted?

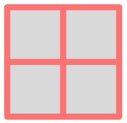




original image

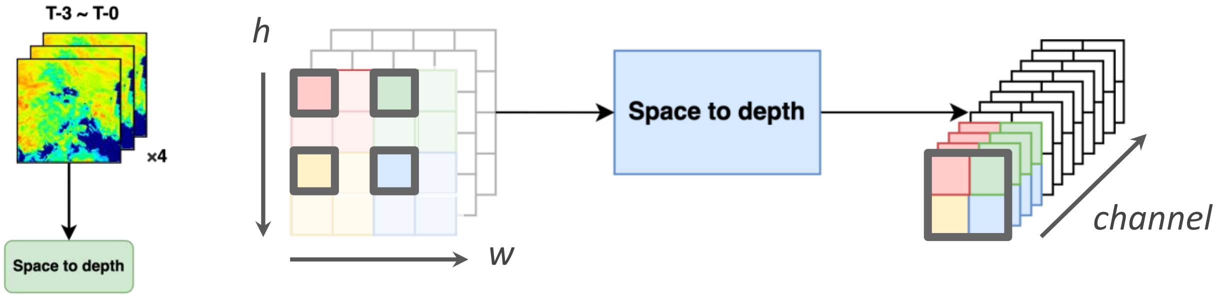
spatial features with 1-5 km scale ←

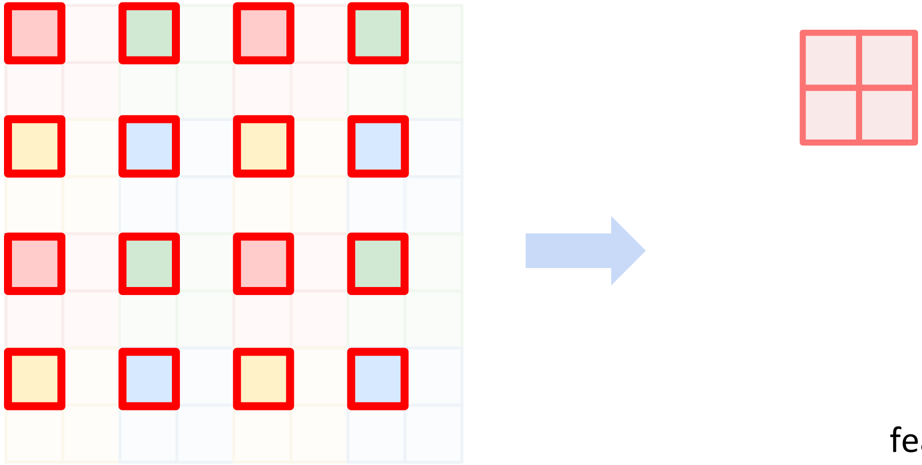
feature map



Unraveling the mystery of DeepMind’s rainfall nowcasting: a step-by-step tutorial for hydrologists

Space to Depth + Convolution: How features are extracted?



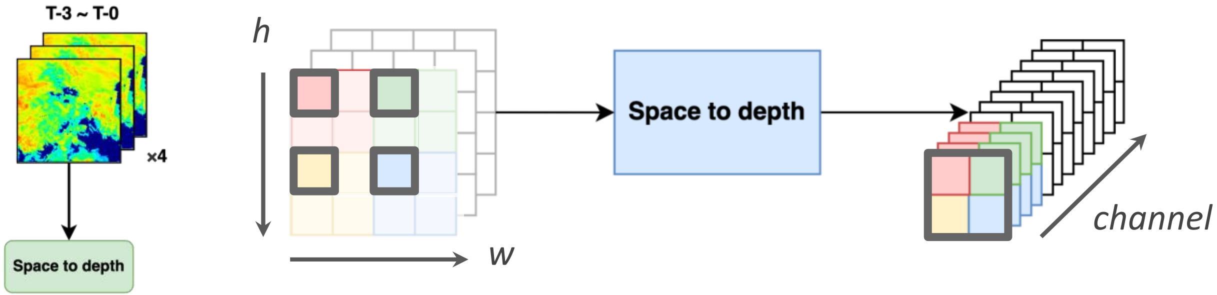


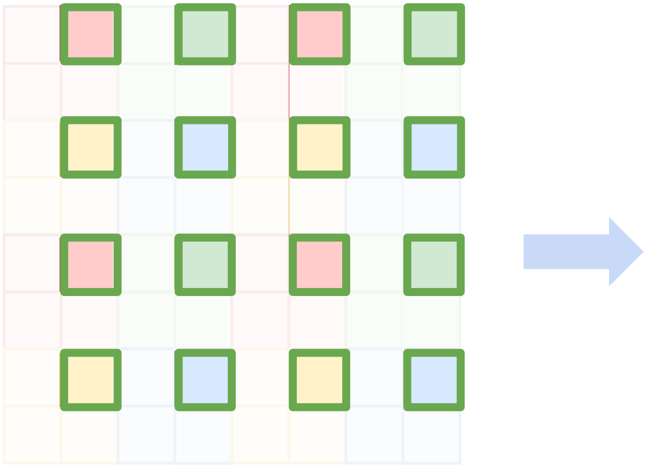
original image

feature maps

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Space to Depth + Convolution: How features are extracted?

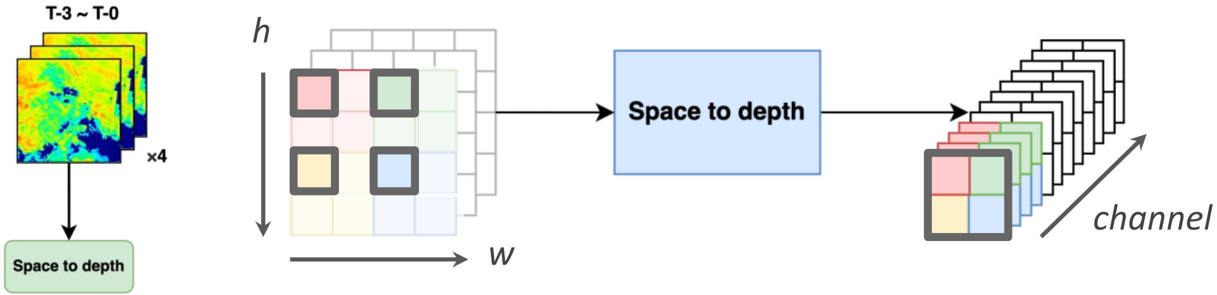


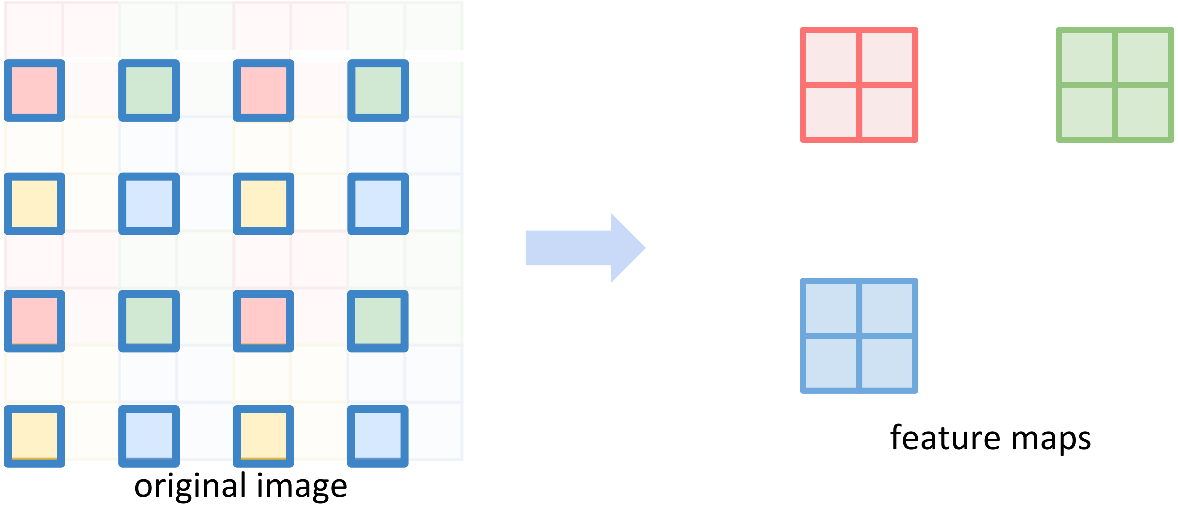


original image

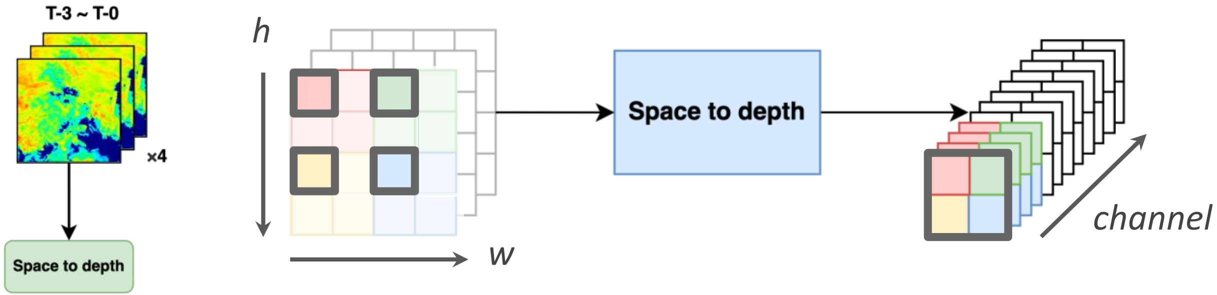
feature maps

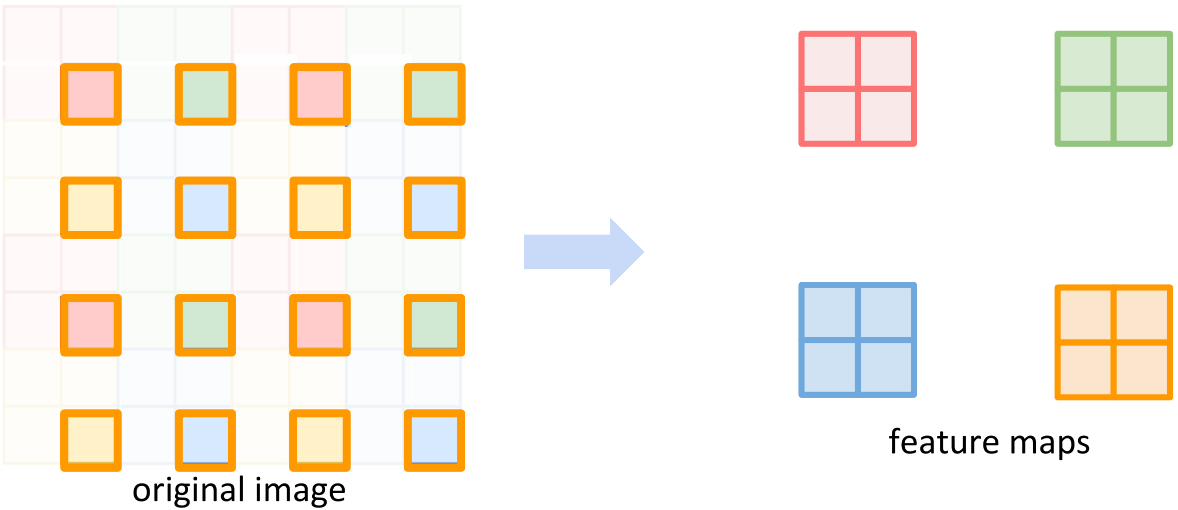
Space to Depth + Convolution: How features are extracted?



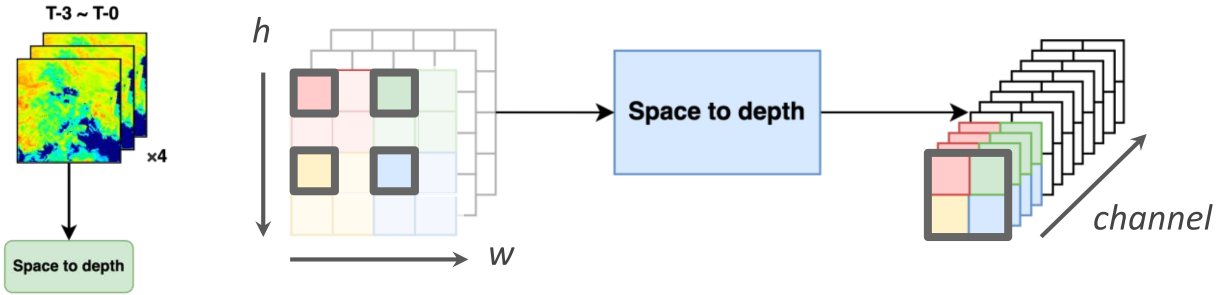


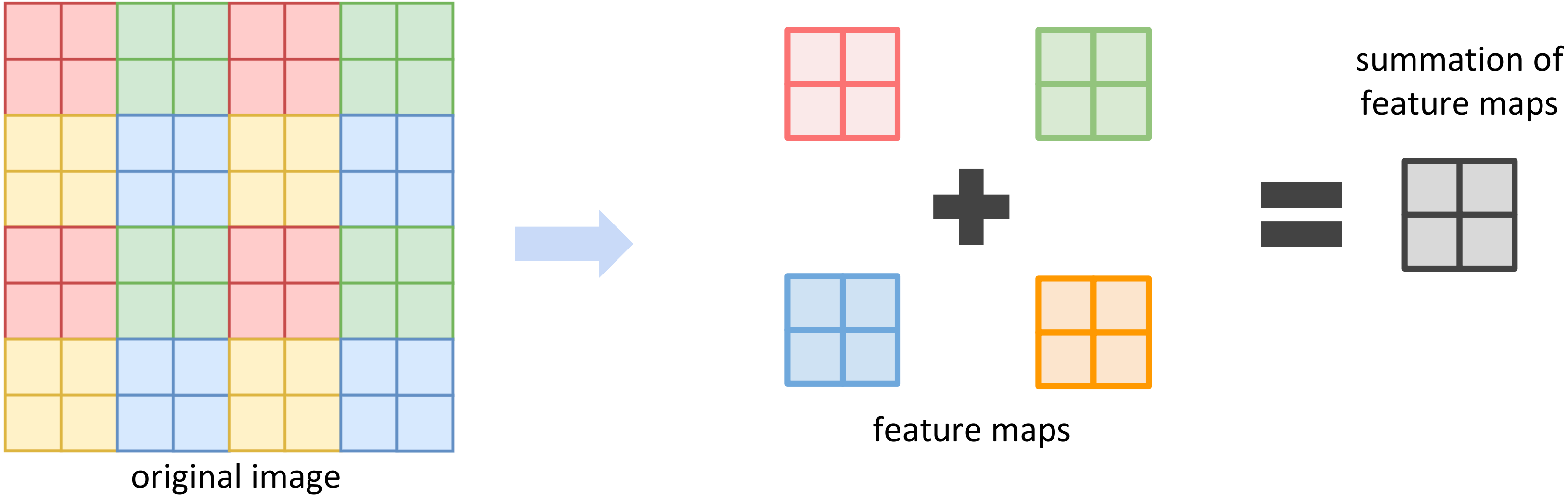
Space to Depth + Convolution: How features are extracted?





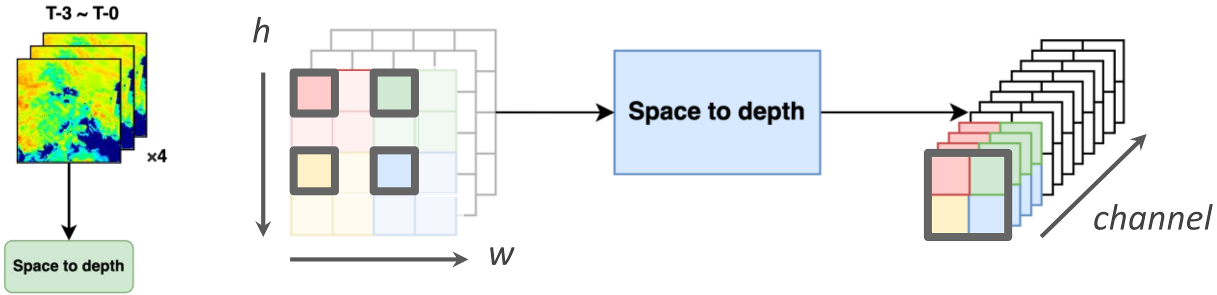
Space to Depth + Convolution: How features are extracted?

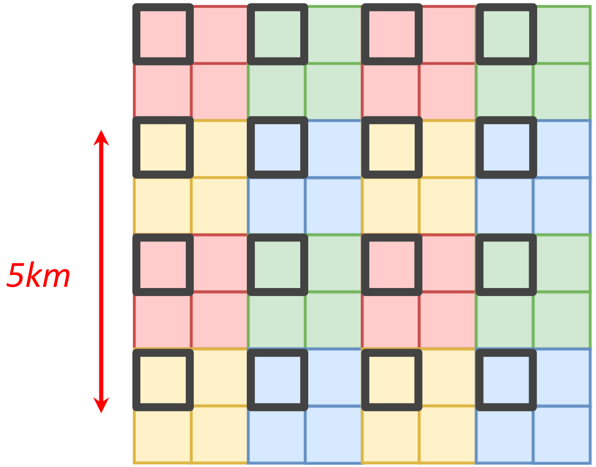




Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Space to Depth + Convolution: How features are extracted?



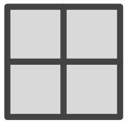


original image

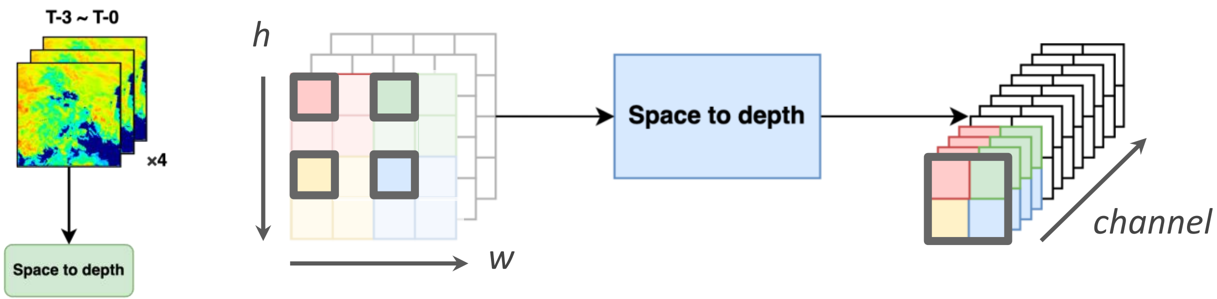
The final feature map contains:

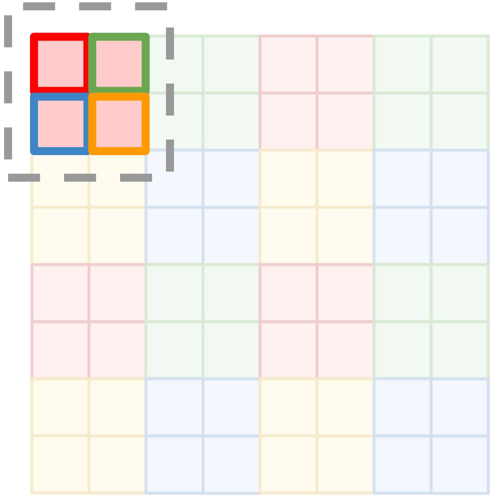
(1) spatial features across 1-5-km scale,

summation of feature maps



Space to Depth + Convolution: How features are extracted?





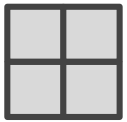
original image

The final feature map contains:

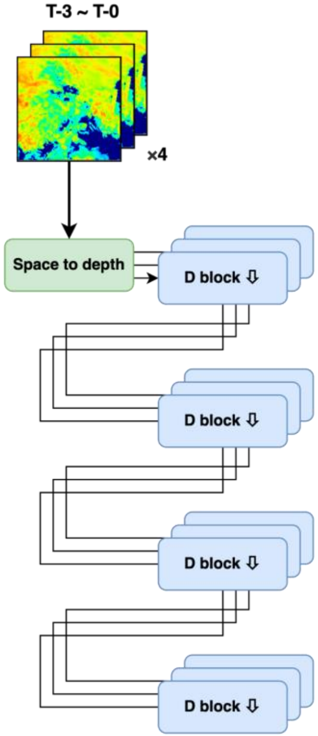
(1) spatial features across 1-5-km scale,

(2) spatial features of neighbouring pixels.

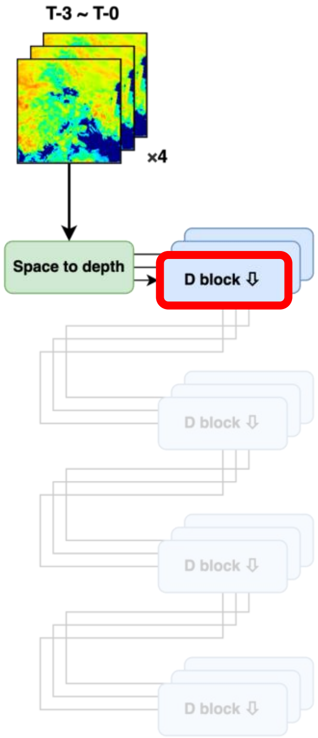
summation of feature maps



Hierachical feature extractor: D Blocks



Hierachical feature extractor: How do D Blocks work?



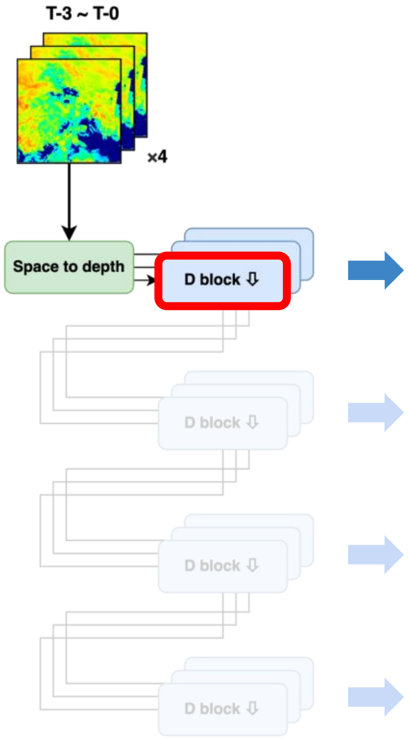
A single D block does operations mentioned below.

(1) 1×1 convolution
(2) 3×3 convolution + 3×3 convolution
(3) 2×2 average pooling

operates in parallel

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

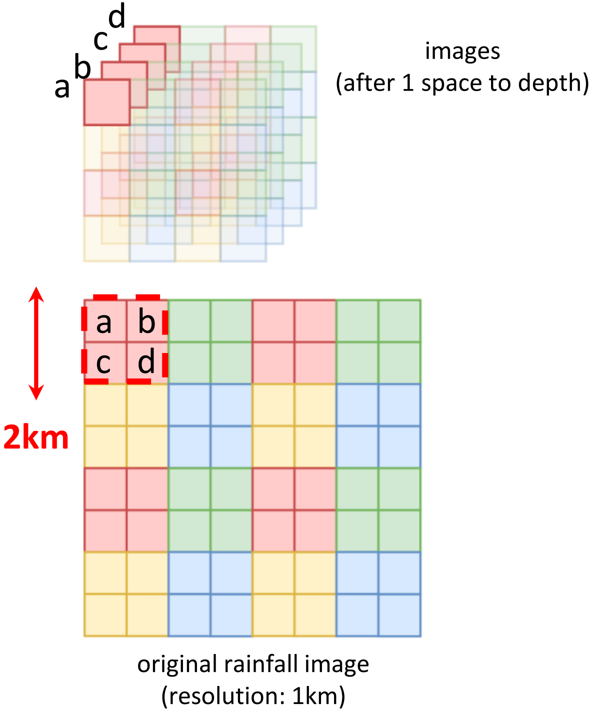
Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

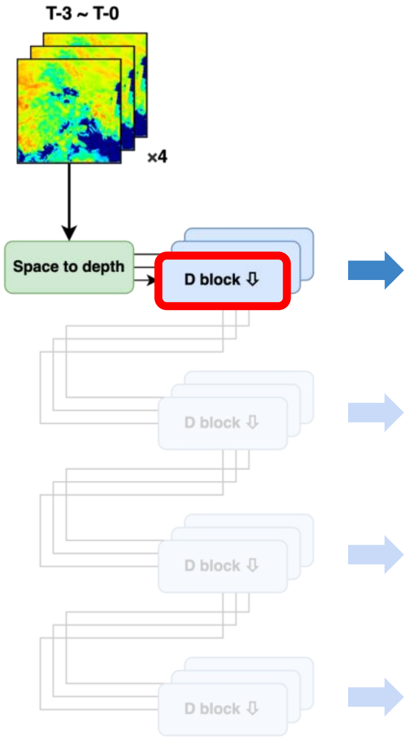
original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	**1st 3×3 convolution**
≦2km × 2km	



Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

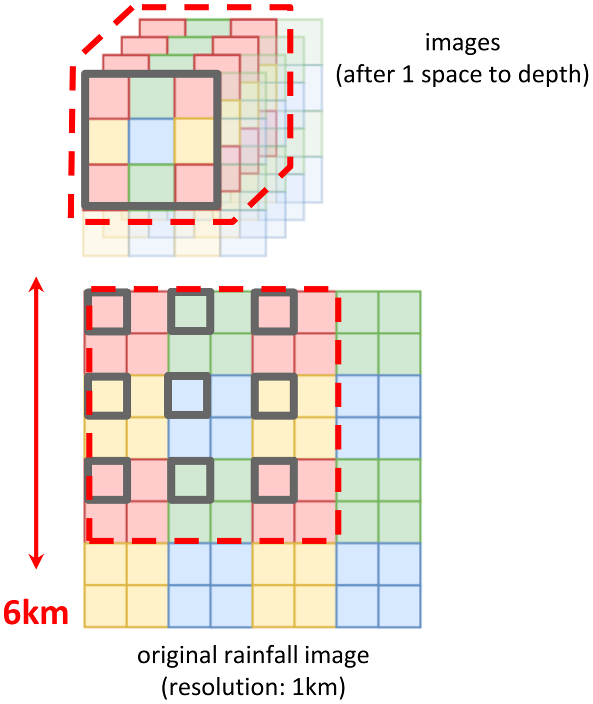
Hierachical feature extractor: Extract features in different levels of spatial extent



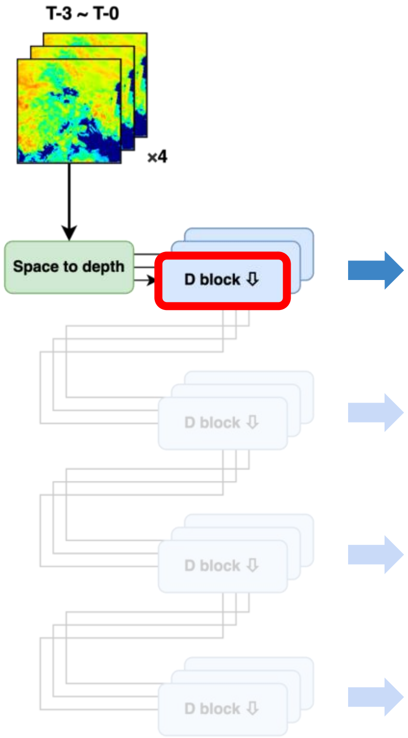
Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	**1st 3×3 convolution**
≦2km × 2km	*≦6km × 6km*



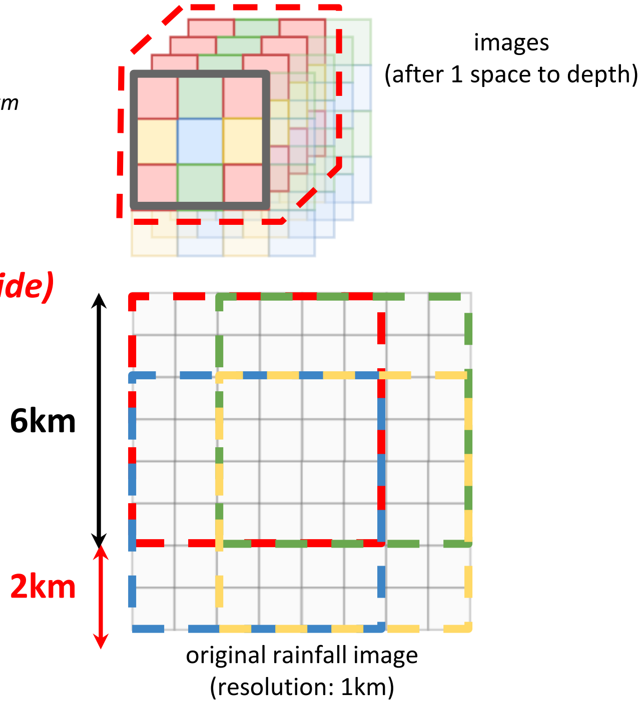
Hierachical feature extractor: Extract features in different levels of spatial extent



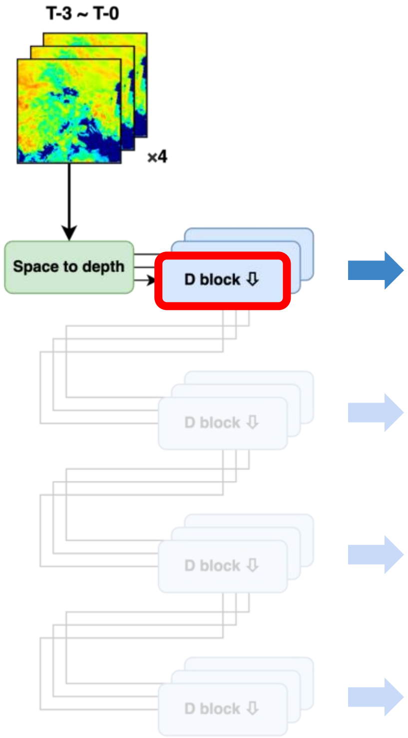
Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1st 3×3 convolution	2nd 3×3 convolution
≦ *6km × 6km*	***6 + 2(stride) + 2(stride)***



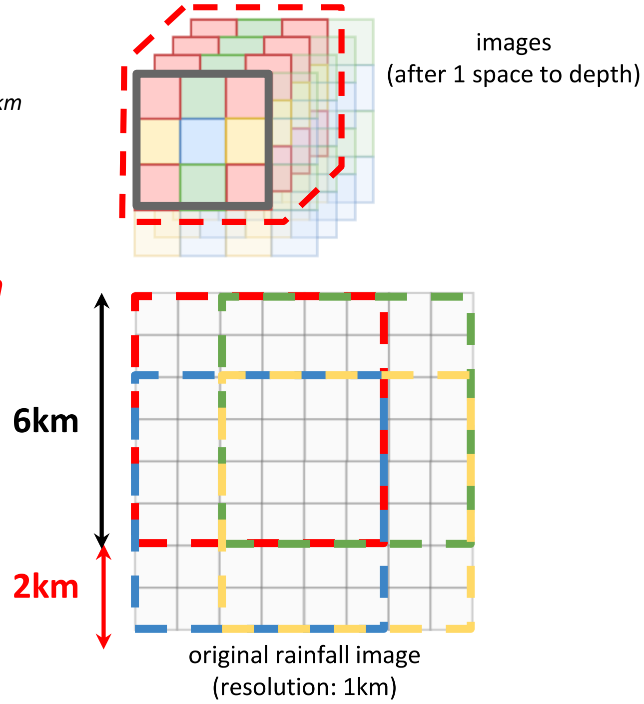
Hierachical feature extractor: Extract features in different levels of spatial extent



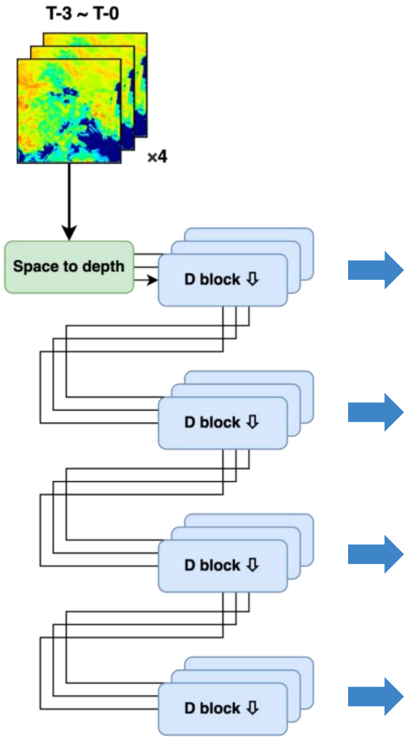
Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1st 3×3 convolution	2nd 3×3 convolution
≦6km × 6km	***≦10km × 10km***



Hierachical feature extractor: Extract features in different levels of spatial extent

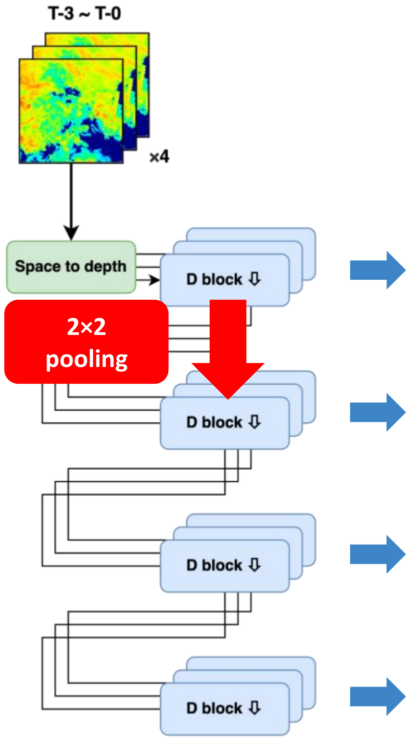


Extracted features’ spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	***2km***	***1km***

Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

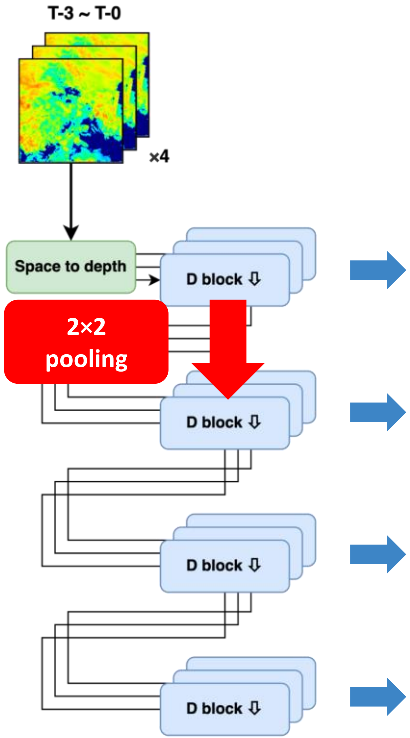
1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*

2×2 pooling

> aggregate cohesive 2×2 pixels into one

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

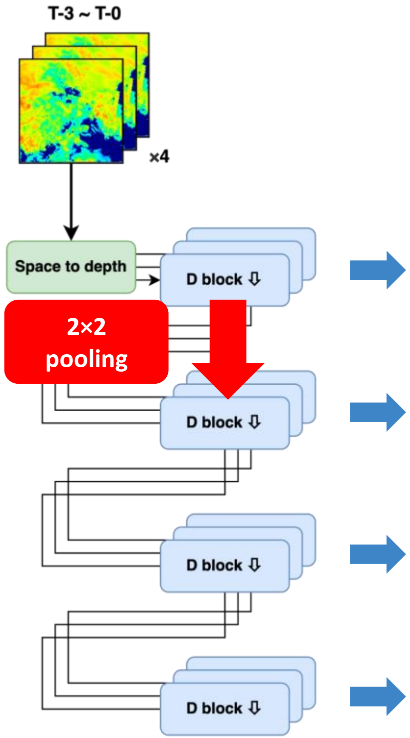
1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	$\boxed{2km}$	*1km*

2×2 pooling

> aggregate cohesive 2×2 pixels into one

> stride length between 2 pixels = 2

Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*

2×2 pooling

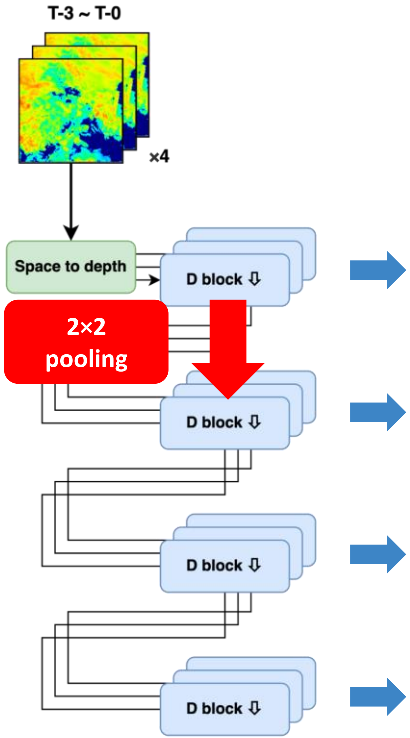
> aggregate cohesive 2×2 pixels into one

> stride length between 2 pixels = 2

> scale of output feature map of 1st D = 10

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*
				12km

2×2 pooling

> aggregate cohesive 2×2 pixels into one

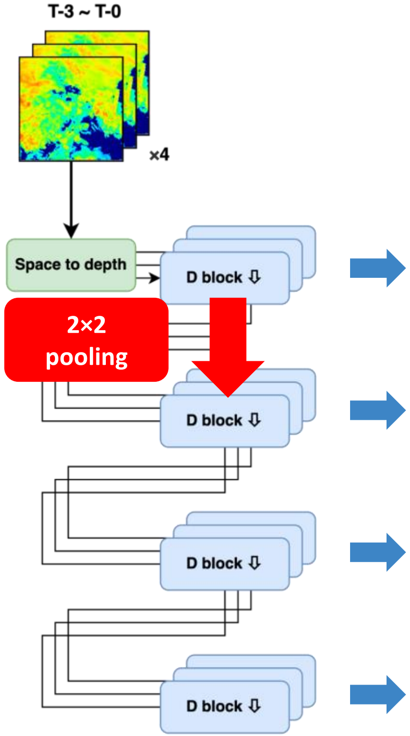
> stride length between 2 pixels = 2

> scale of output feature map of 1st D = 10

> scale of input of 2nd D = 10 + 2(one stride)

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Hierachical feature extractor: Extract features in different levels of spatial extent

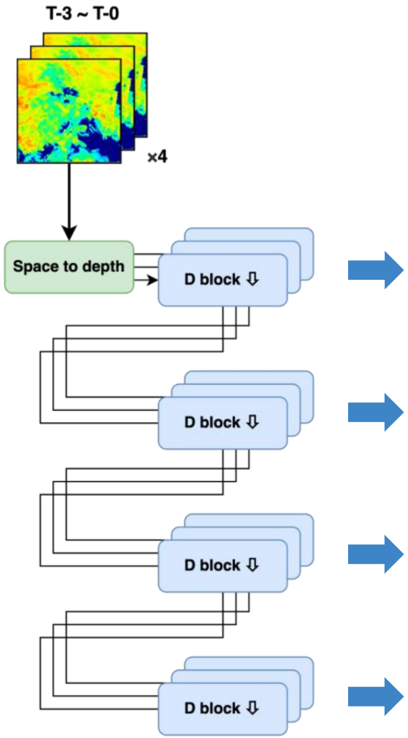


Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*
			×2	
			4km	*12km*

Hierachical feature extractor: Extract features in different levels of spatial extent

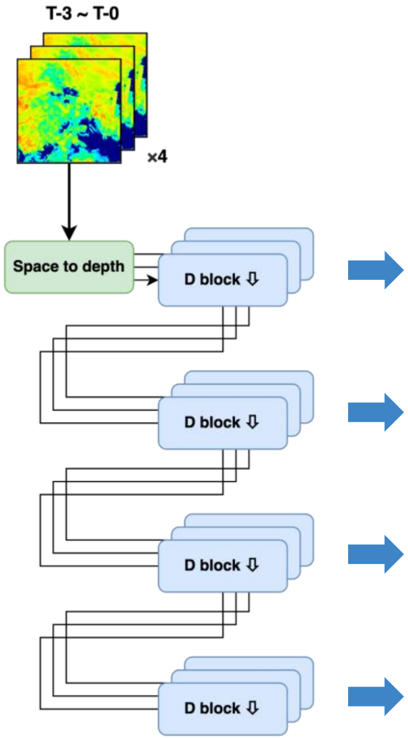


Extracted features’ spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*
≦12km × 12km	***12 + 4 + 4*** ***≦20km × 20km***	***20 + 4 + 4*** ***≦28km × 28km***	*4km*	*12km*

Hierachical feature extractor: Extract features in different levels of spatial extent

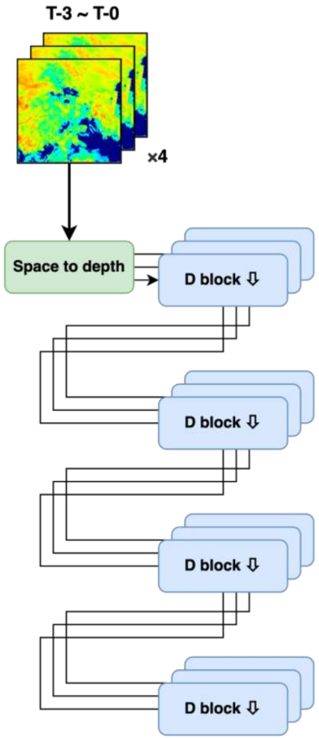


Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*
≦12km × 12km	*12 + 4 + 4* *≦20km × 20km*	*20 + 4 + 4* *≦28km × 28km*	*4km*	*12km*
≦32km × 32km	***32 + 8 + 8*** ***≦48km × 48km***	***48 + 8 + 8*** ***≦64km × 64km***	*8km*	***28 + 4*** *32km*

Hierachical feature extractor: Extract features in different levels of spatial extent



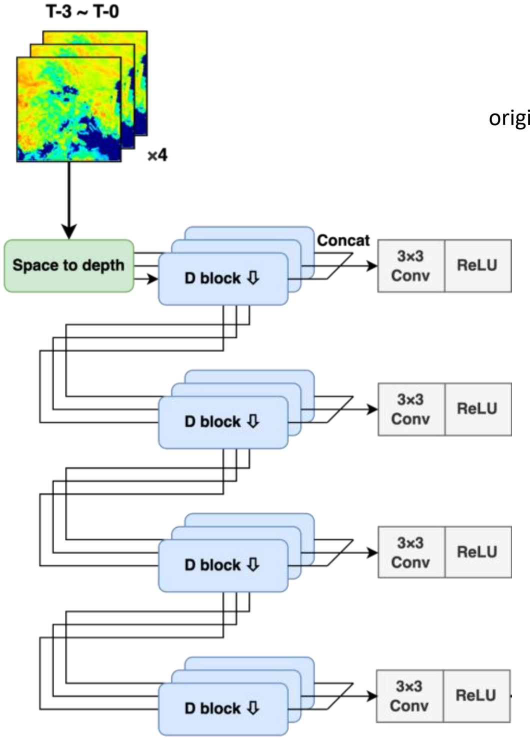
Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

1×1 convolution	1st 3×3 convolution	2nd 3×3 convolution	stride length	side length of pixel
≦2km × 2km	*≦6km × 6km*	*≦10km × 10km*	*2km*	*1km*
≦12km × 12km	*12 + 4 + 4* *≦20km × 20km*	*20 + 4 + 4* *≦28km × 28km*	*4km*	*12km*
≦32km × 32km	*32 + 8 + 8* *≦48km × 48km*	*48 + 8 + 8* *≦64km × 64km*	*8km*	*32km*
≦72km × 72km	***72 + 16 + 16*** ***≦104km × 104km***	***104 + 16 + 16*** ***≦136km × 136km***	*16km*	***64 + 8*** ***72km***

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

Hierachical feature extractor: Extract features in different levels of spatial extent



Extracted features' spatial extent

original image size = *256km × 256km*, resolution = *1km × 1km*

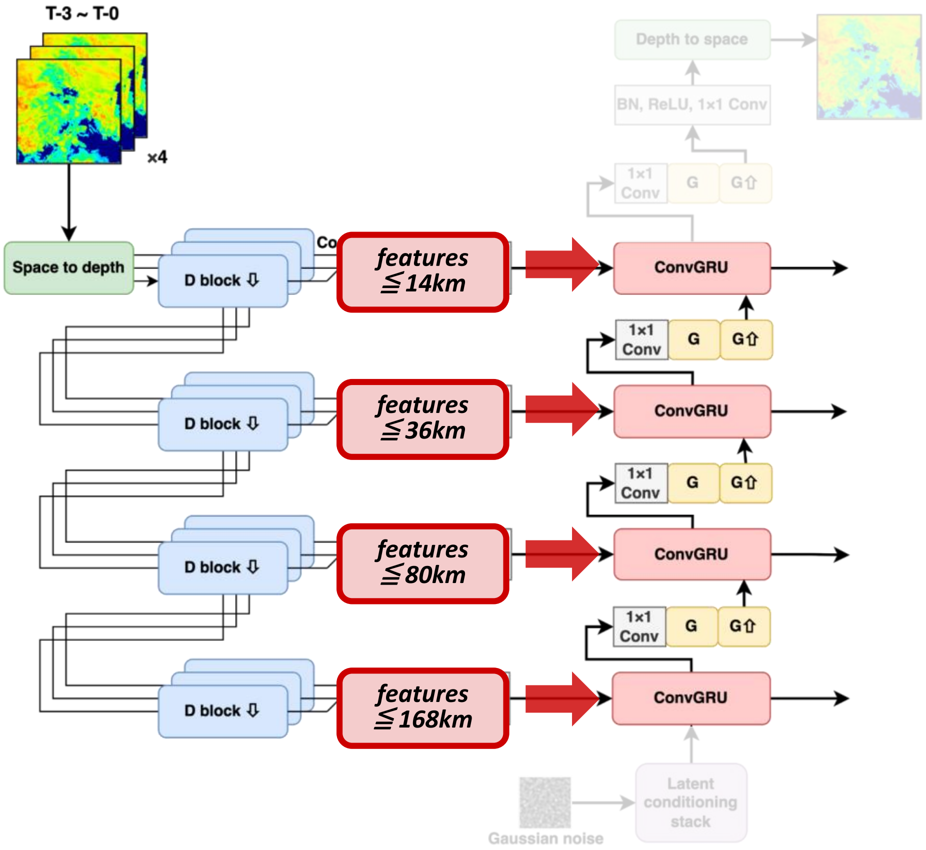
≦14km × 14km

≦36km × 36km

≦80km × 80km

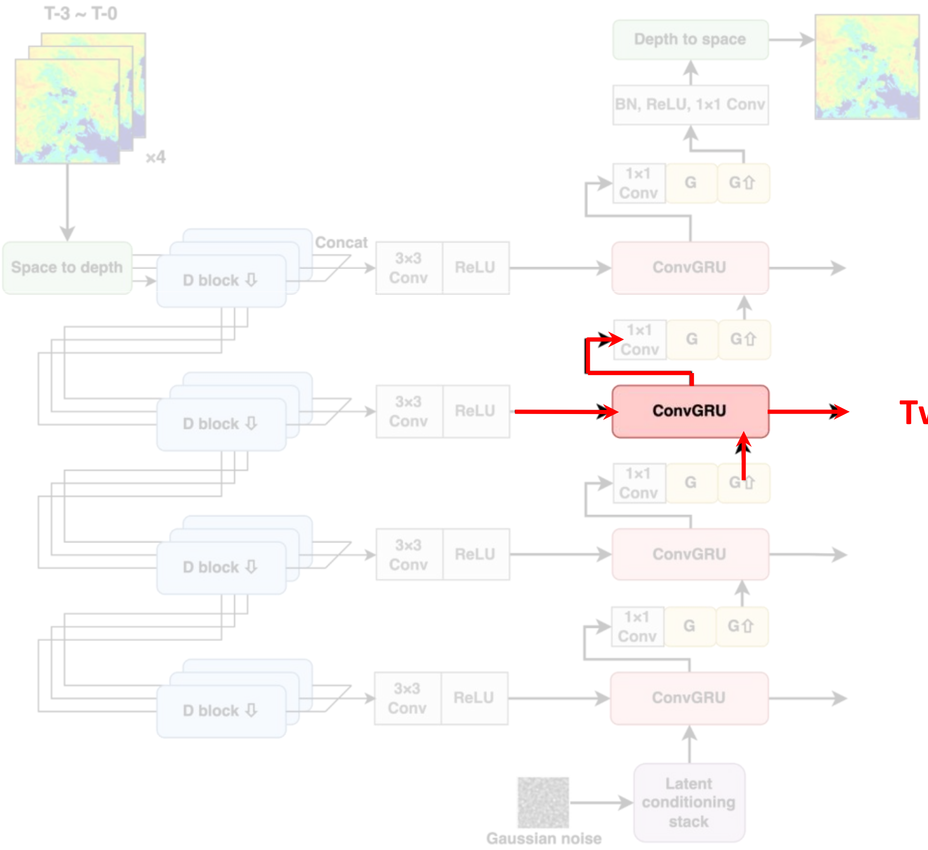
≦168km × 168km

ConvGRU


T-3 ~ T-0
×4
Space to depth
D block ⇩
D block ⇩
D block ⇩
D block ⇩
features ≦14km
features ≦36km
features ≦80km
features ≦168km
ConvGRU
ConvGRU
ConvGRU
ConvGRU
1×1 Conv
G
G⇧
Depth to space
BN, ReLU, 1×1 Conv
Latent conditioning stack
Gaussian noise

Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

ConvGRU



Two inputs and two outputs.

ConvGRU: How does it work?

Components of Gated Recurrent Unit (GRU)

Inputs	Operations	Factors	Outputs
Hidden state	Update gate	Update ratio	New hidden state
Current state	Reset gate	Reset ratio	Output (or new current state)
	Output gate		

ConvGRU: How does it work?

Components of Gated Recurrent Unit (GRU)

Inputs	Operations	Factors	Outputs
Hidden state	Update gate	Update ratio	New hidden state
Current state	Reset gate	Reset ratio	Output (or new current state)
	Output gate		

Each gate contains a convolution.

ConvGRU: How does it work?

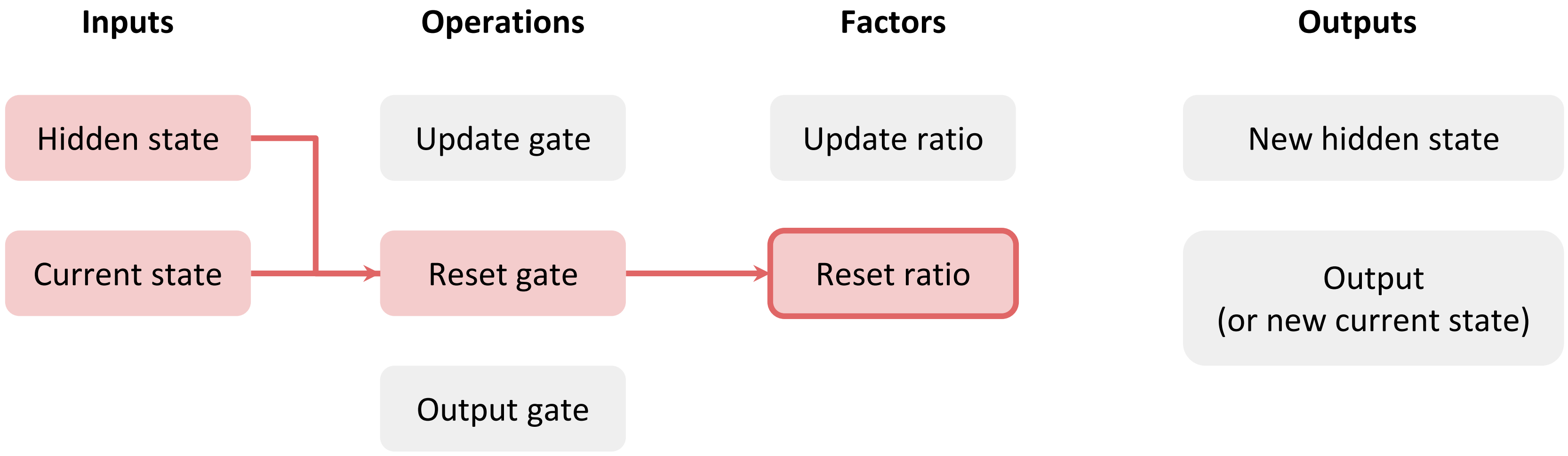
Components of Gated Recurrent Unit (GRU)

Inputs	Operations	Factors	Outputs
Hidden state	Update gate	Update ratio	New hidden state
Current state	Reset gate	Reset ratio	Output (or new current state)
	Output gate		

- ***Hidden state*** contains features **accumulated from <u>first input frame</u> to <u>input frame before the target time step</u>**. → **long-term impact**
- ***Current state*** contains features from **input frame <u>before the target time step only</u>**.
  → **immediate impact**

ConvGRU: How does it work?

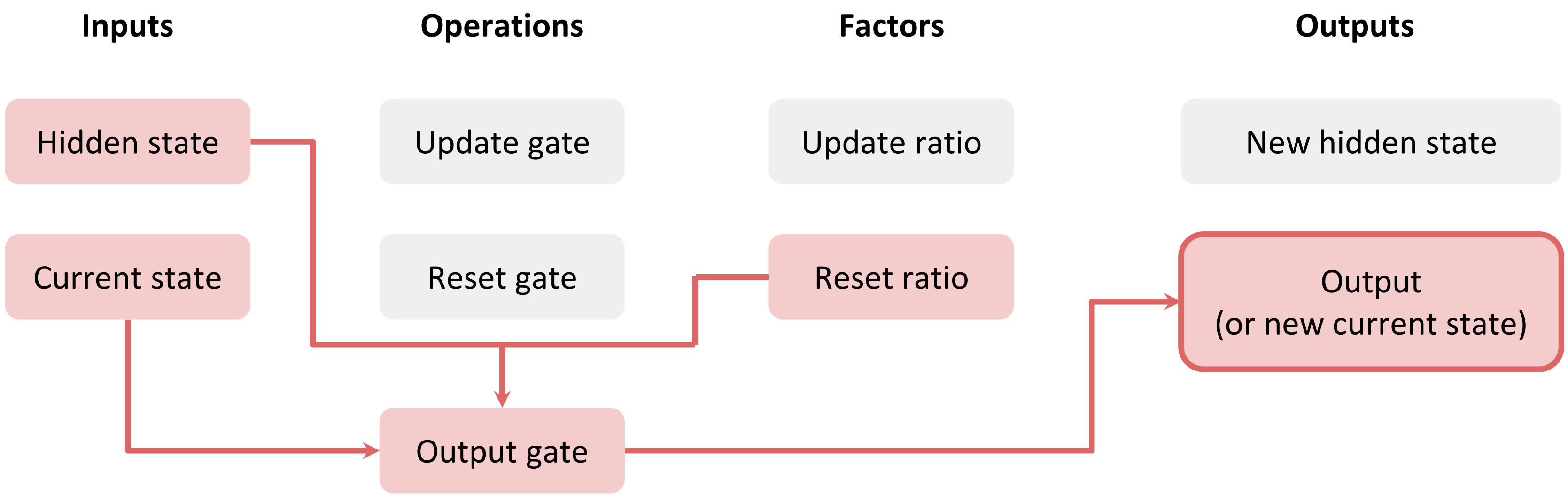
Components of Gated Recurrent Unit (GRU)



Reset ratio is determined by *<u>hidden state</u>*, *<u>current state</u>*, and *<u>weights of reset gate</u>*.

ConvGRU: How does it work?

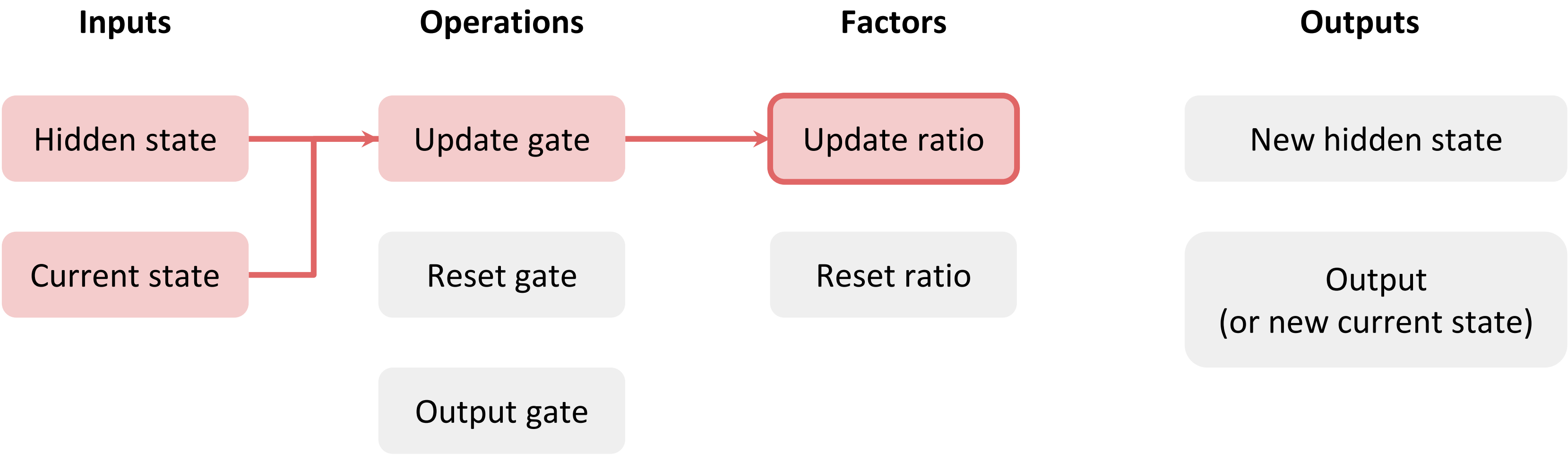
Components of Gated Recurrent Unit (GRU)



Output is determined by *hidden state × reset ratio*, *current state*, and *weights of output gate*.

ConvGRU: How does it work?

Components of Gated Recurrent Unit (GRU)



Update ratio is determined by *hidden state*, *current state*, and *weights of update gate*.

ConvGRU: How does it work?

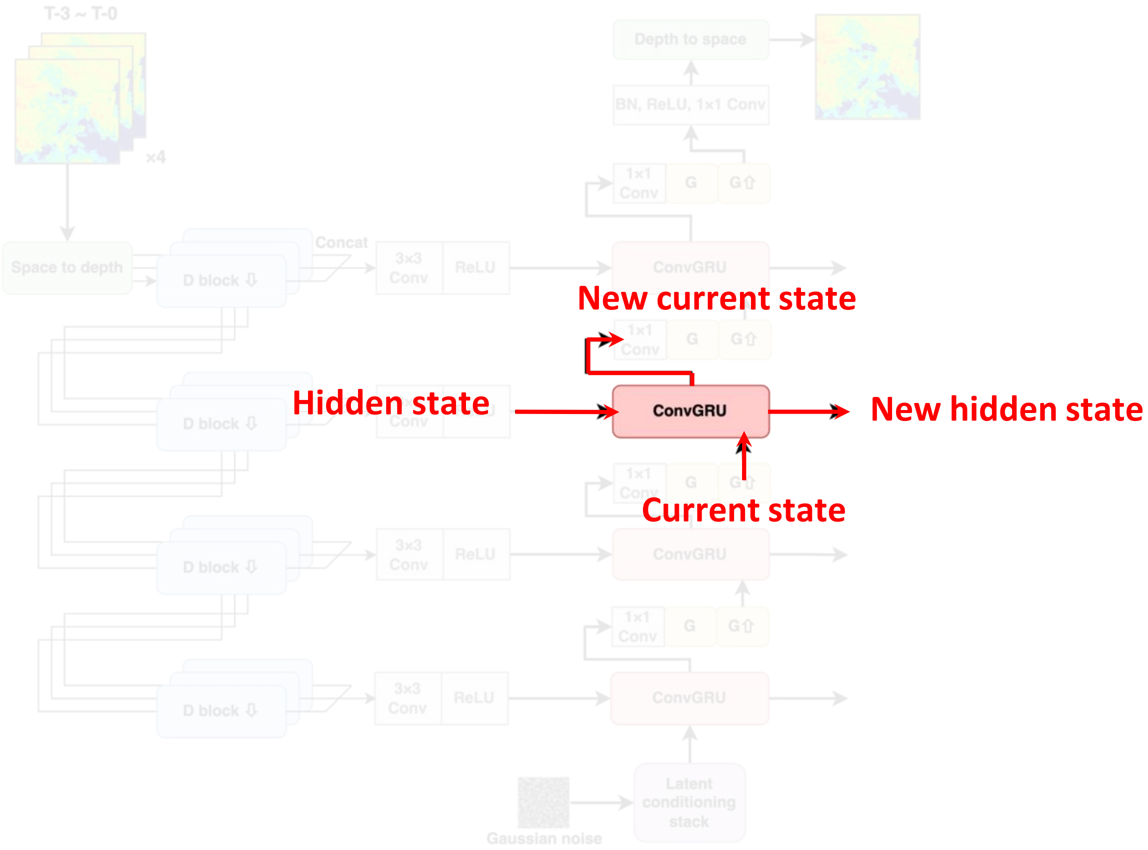
Components of Gated Recurrent Unit (GRU)

Inputs	Operations	Factors	Outputs
Hidden state	Update gate	Update ratio	New hidden state
Current state	Reset gate	Reset ratio	Output (or new current state)
	Output gate		

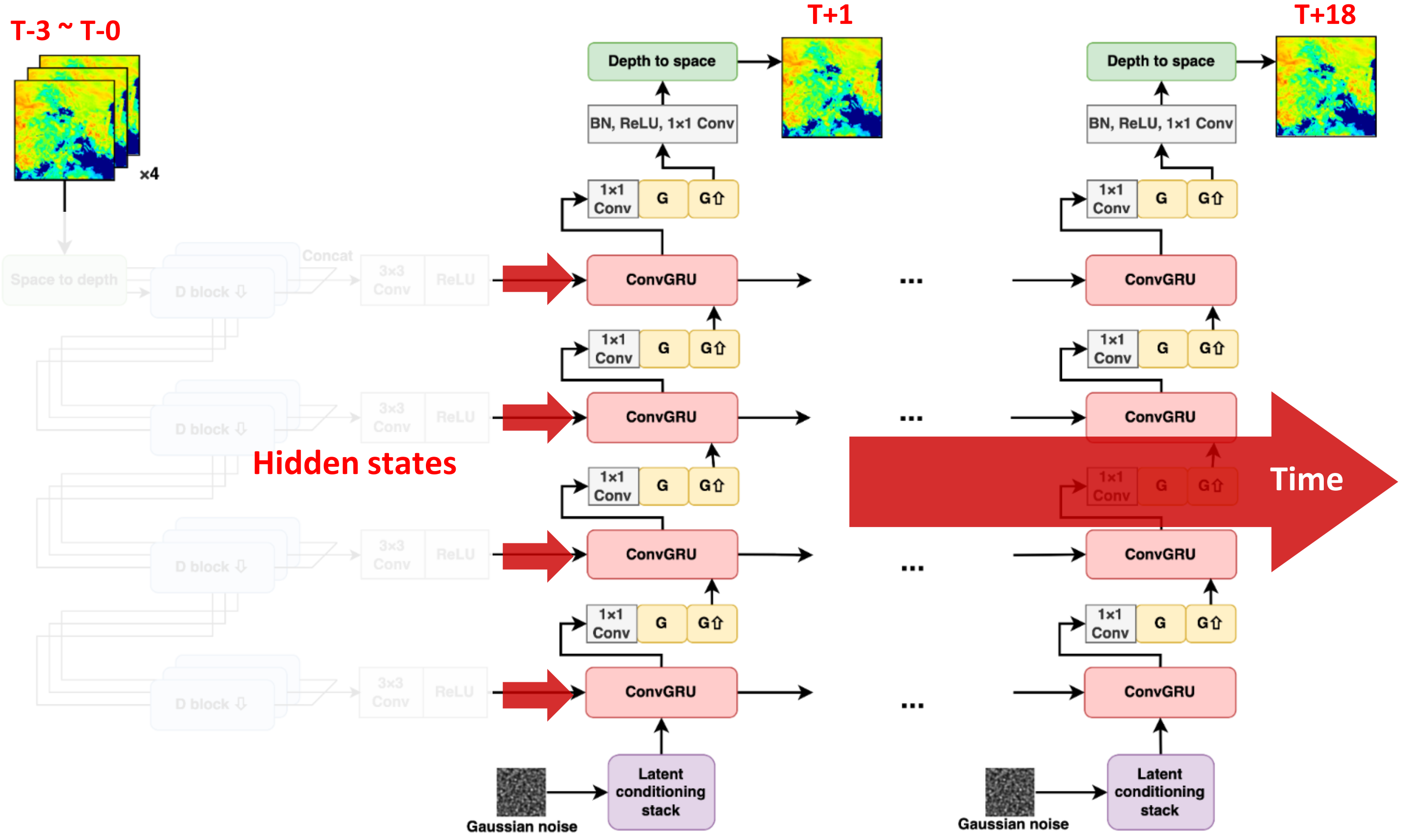
New hidden state is calculated by ***hidden state***, ***update ratio***, and ***output***.

(*New hidden state* = *hidden state* × (1 - *update ratio*) + *output* × *update ratio*)

ConvGRU: The role in DGMR

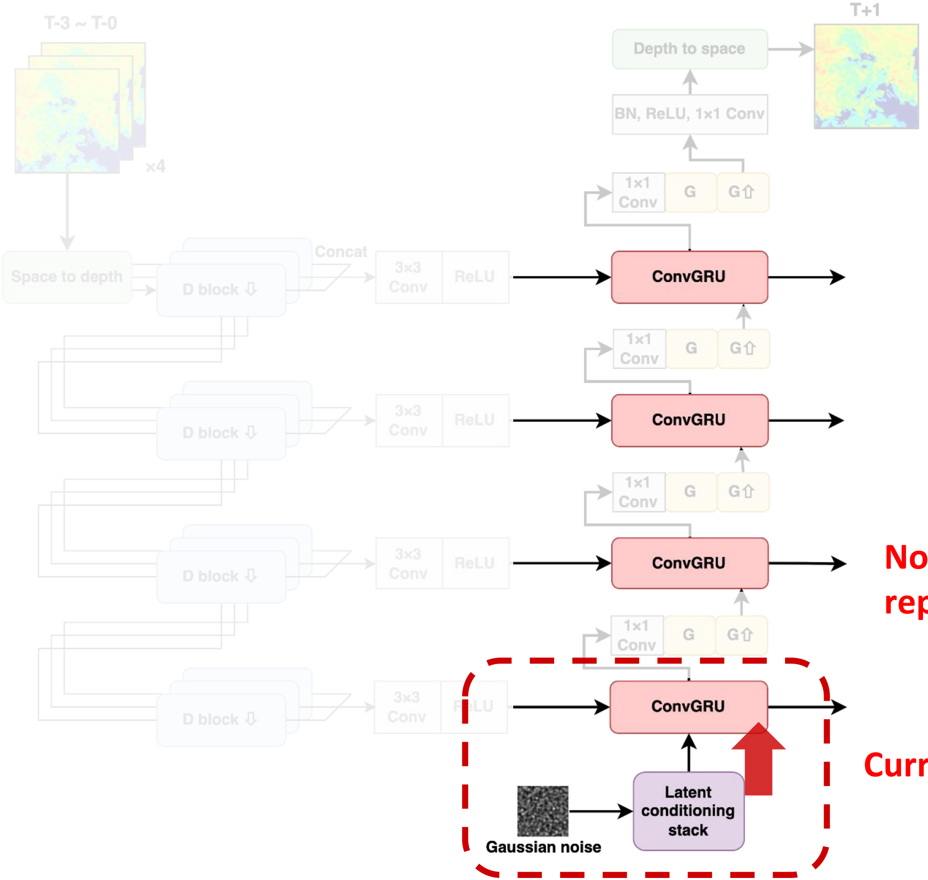


ConvGRU: Temporal feature extractor



Unraveling the mystery of DeepMind's rainfall nowcasting: a step-by-step tutorial for hydrologists

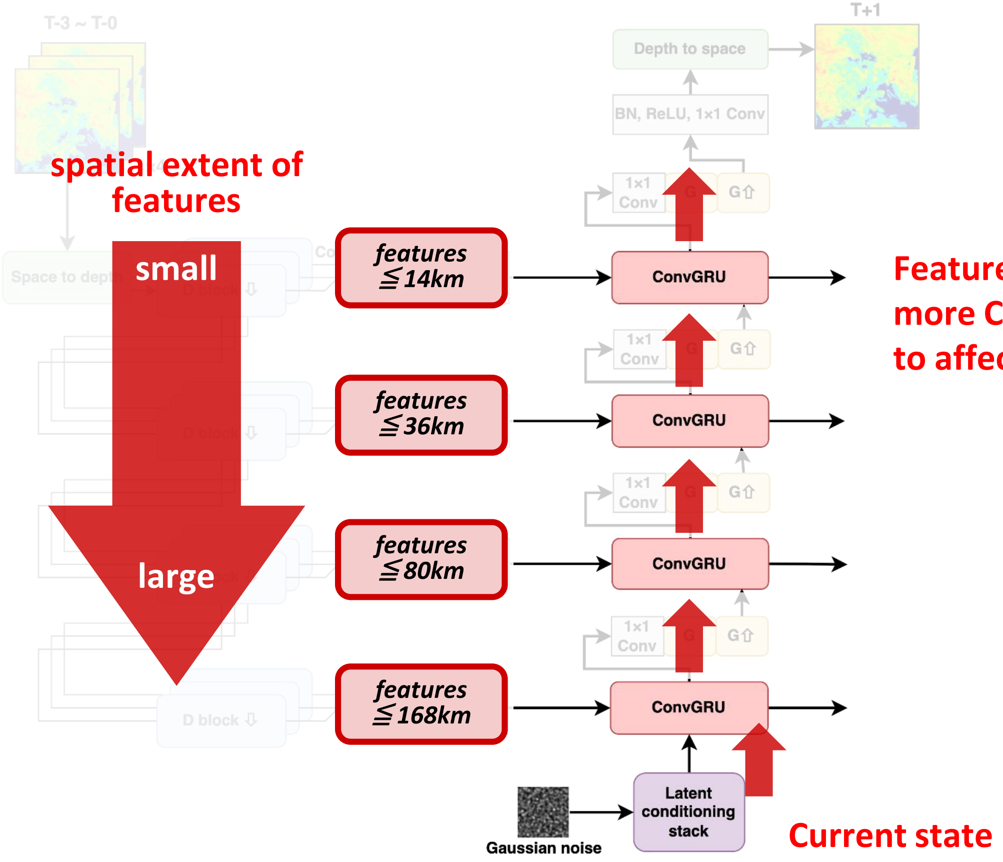
ConvGRU: Spatio-temporal feature extractor



Noise is considered as current state, representing immediate impact on the output.

Current state

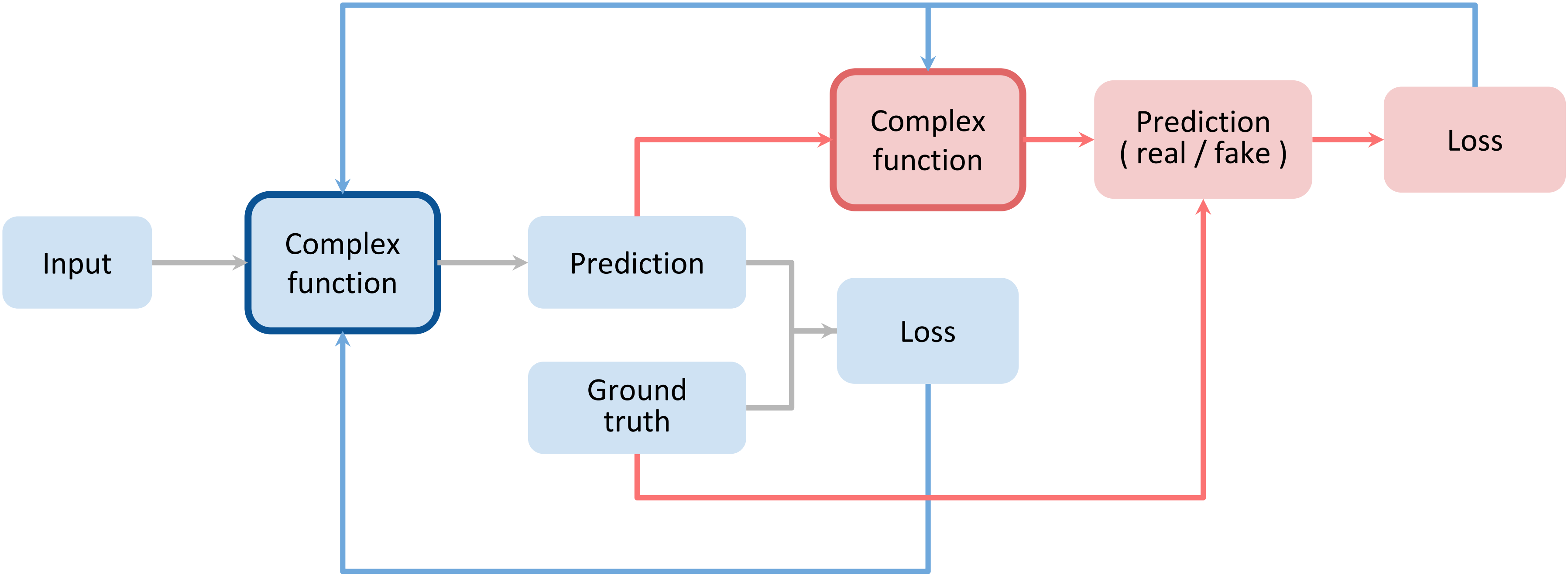
ConvGRU: Spatio-temporal feature extractor



Features at larger spatial extent pass through more ConvGRUs, enabling large spatial features to affect small spatial features.

GAN: Generative Adversarial Network

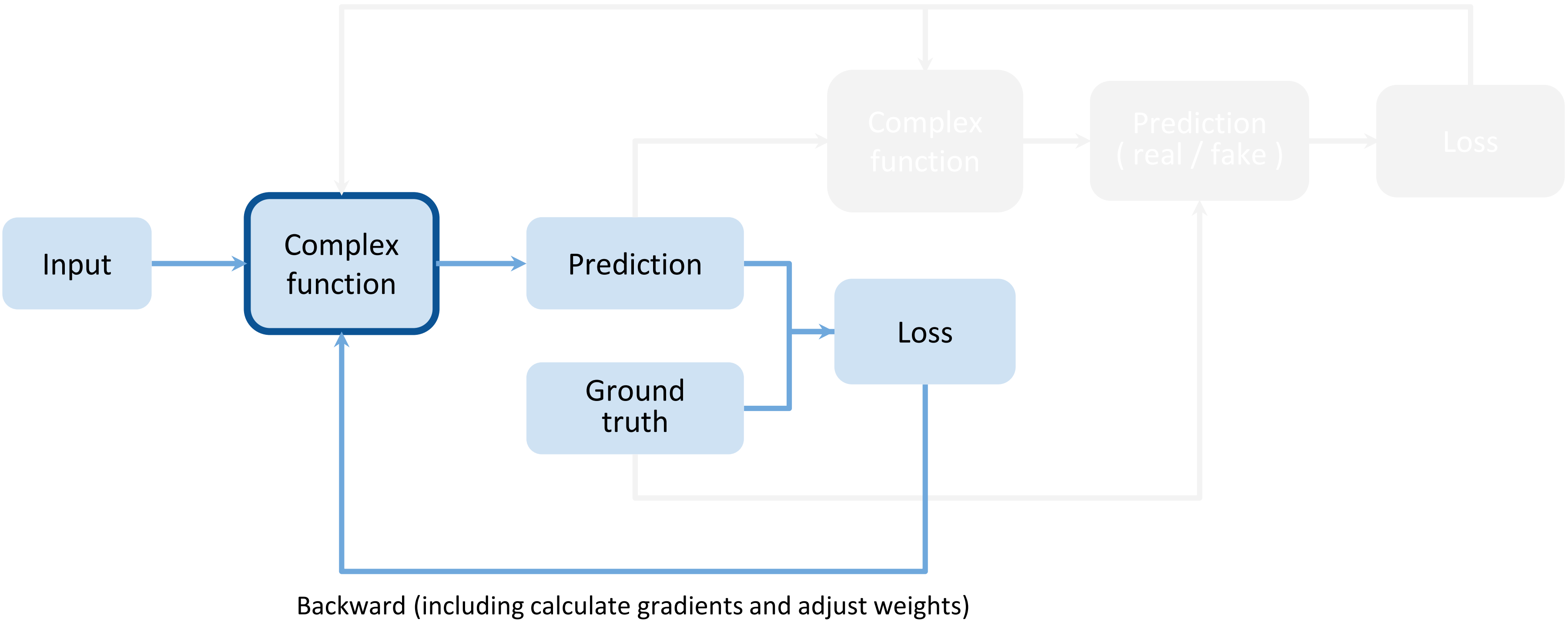
GAN uses another complex function to describe the distance between prediction and ground truth.



Backward (including calculate gradients and adjust weights)

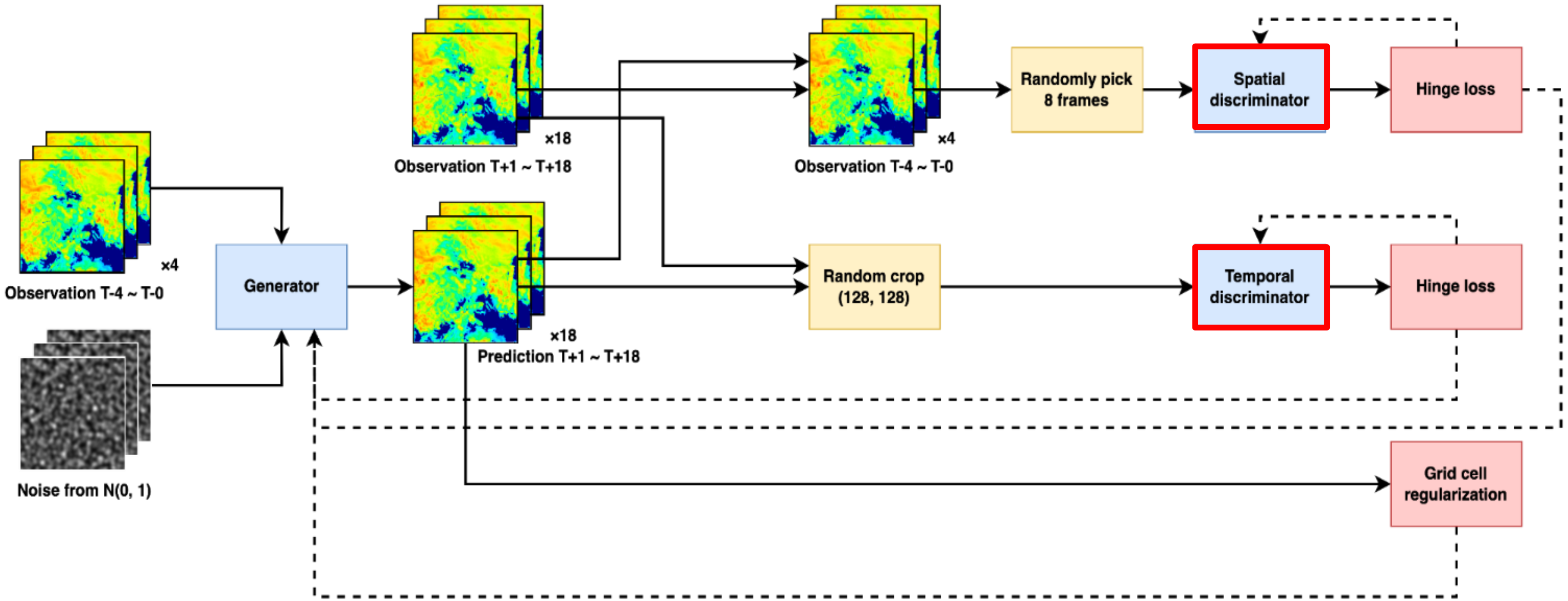
GAN: Generative Adversarial Network

GAN uses another complex function to describe the distance between prediction and ground truth.



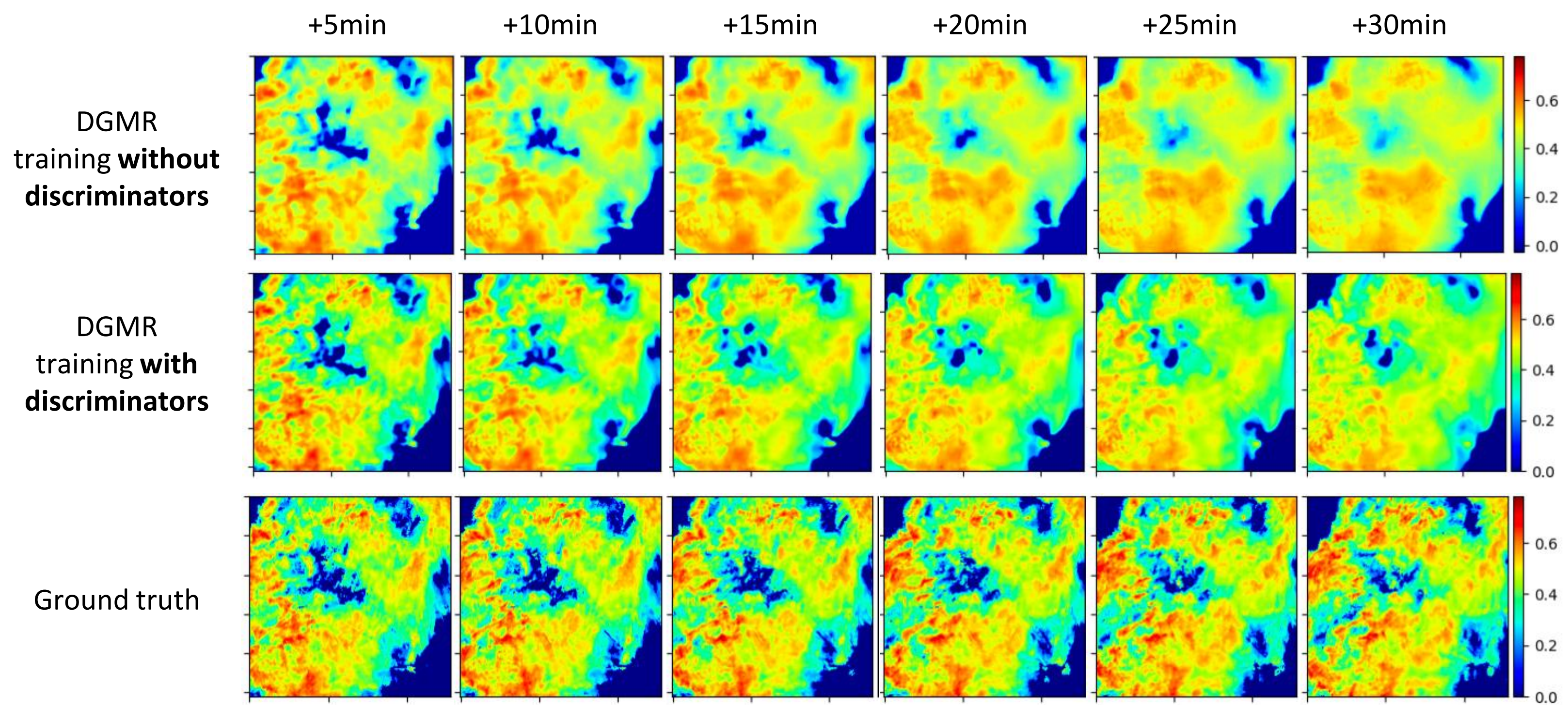
GAN: Framework of DGMR

DGMR uses two discriminators to train their generator.



GAN: The benefit of using GAN

More details can be perserved in the predicted fields when GAN is used.

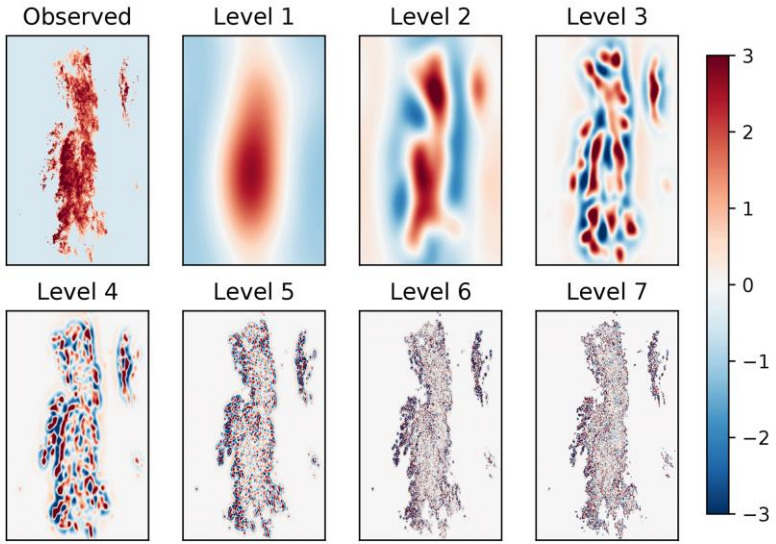


Some insights gained from reproducing DGMR

- Variations in rainfall = Advection + Evolution (in time)
- Spatial and temporal features of rainfall are not independent from each other
- Scale matters!
  - Levels of D blocks, associating with stacks of ConvGRU models
  - Space-to-depth (S2D)
  - Adversarial framework (GAN)

Stay tuned with our developments

Is there a more efficient way to learn spatial and temporal features across scales?



Our DGMR replica on Github!

Yun-Ting Heh (hehyunting@caece.net)
Li-Pen Wang (lpwang@ntu.edu.tw)