Imperial Data Science and AI Winter School 2025

Barin-Tumor Segmentation

Group 3

Li Xiangyu   Su Yi   Wu Kunzhen

PART 01

nnU-Net Introduction

nnU-Net

nnU-Net

a self-configuring method for deep learning-based biomedical image segmentation

Detailed Configuration

Detailed configuration is more important than architectural design. nnU-Net uses a simple U-Net architecture, but transcends more complex parameters through good configuration.

Automated Configuration

Systemize the complex manual method configuration process into fixed parameters, rule-based parameters and empirical parameters.

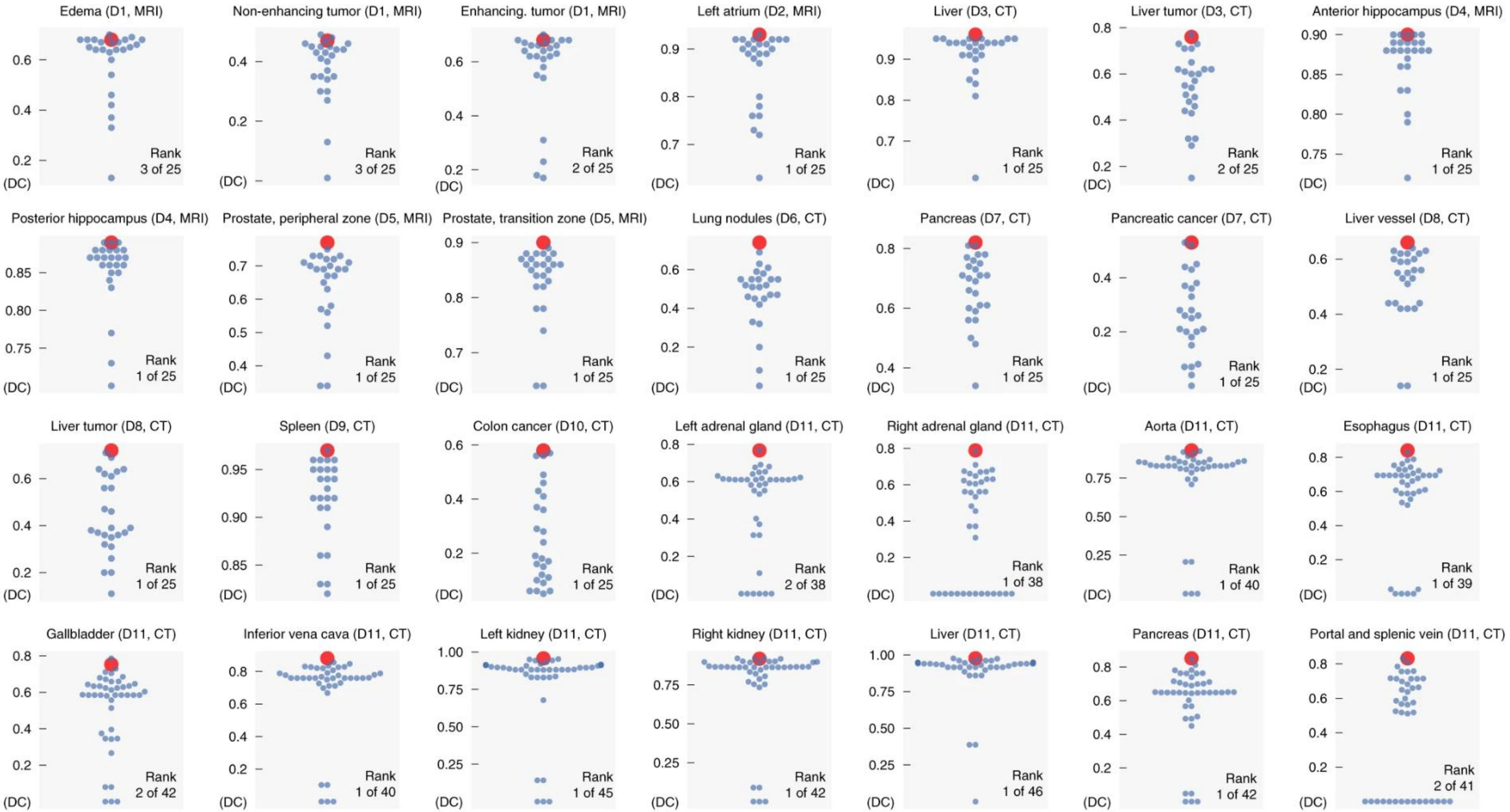
Adaptable Configuration

Twenty-three publicly available biomedical image datasets were used in the development and evaluation of nnU-Net.

Easy Configuration

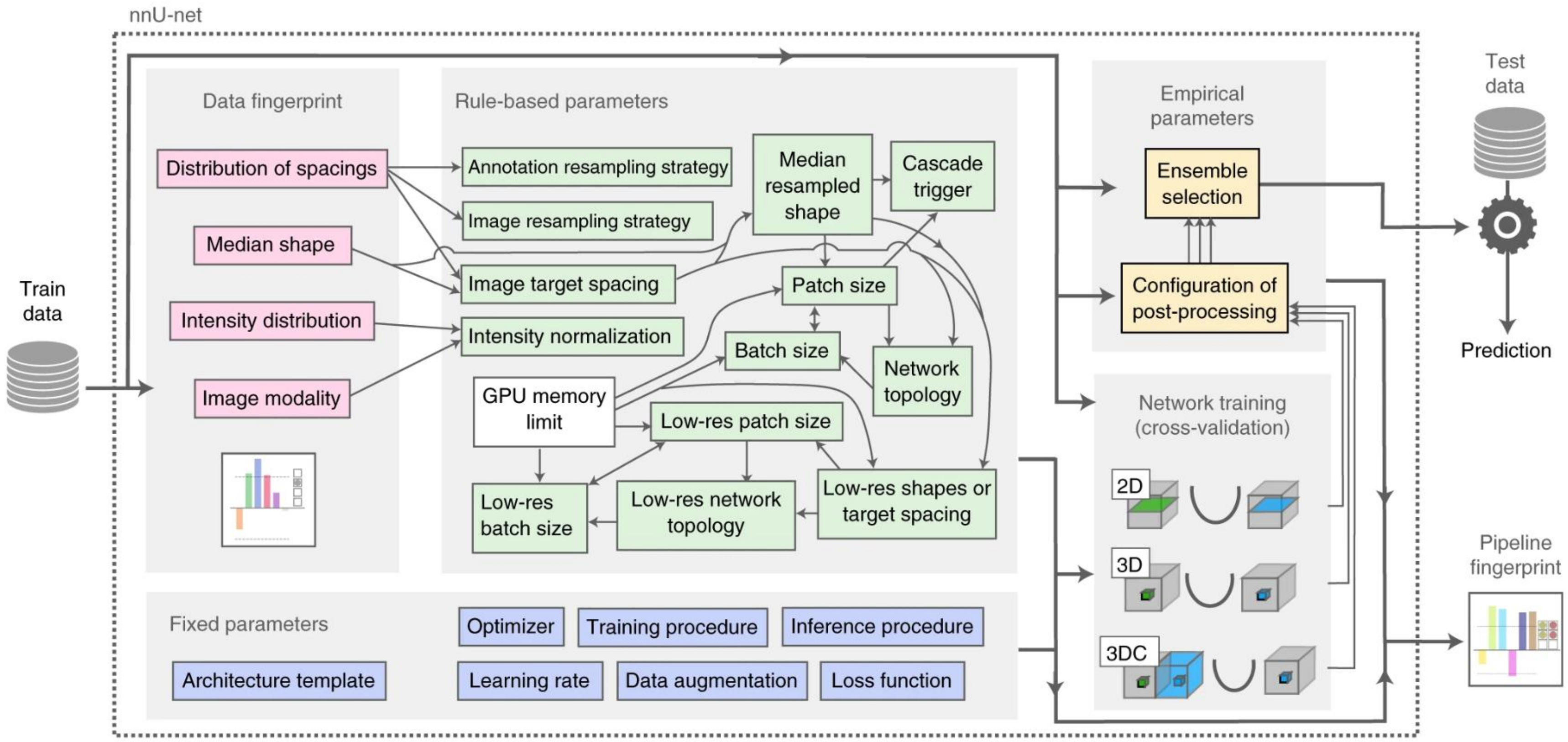
As an open source tool, nnU-Net can be used immediately by people who are not familiar with deep learning.

nnU-Net

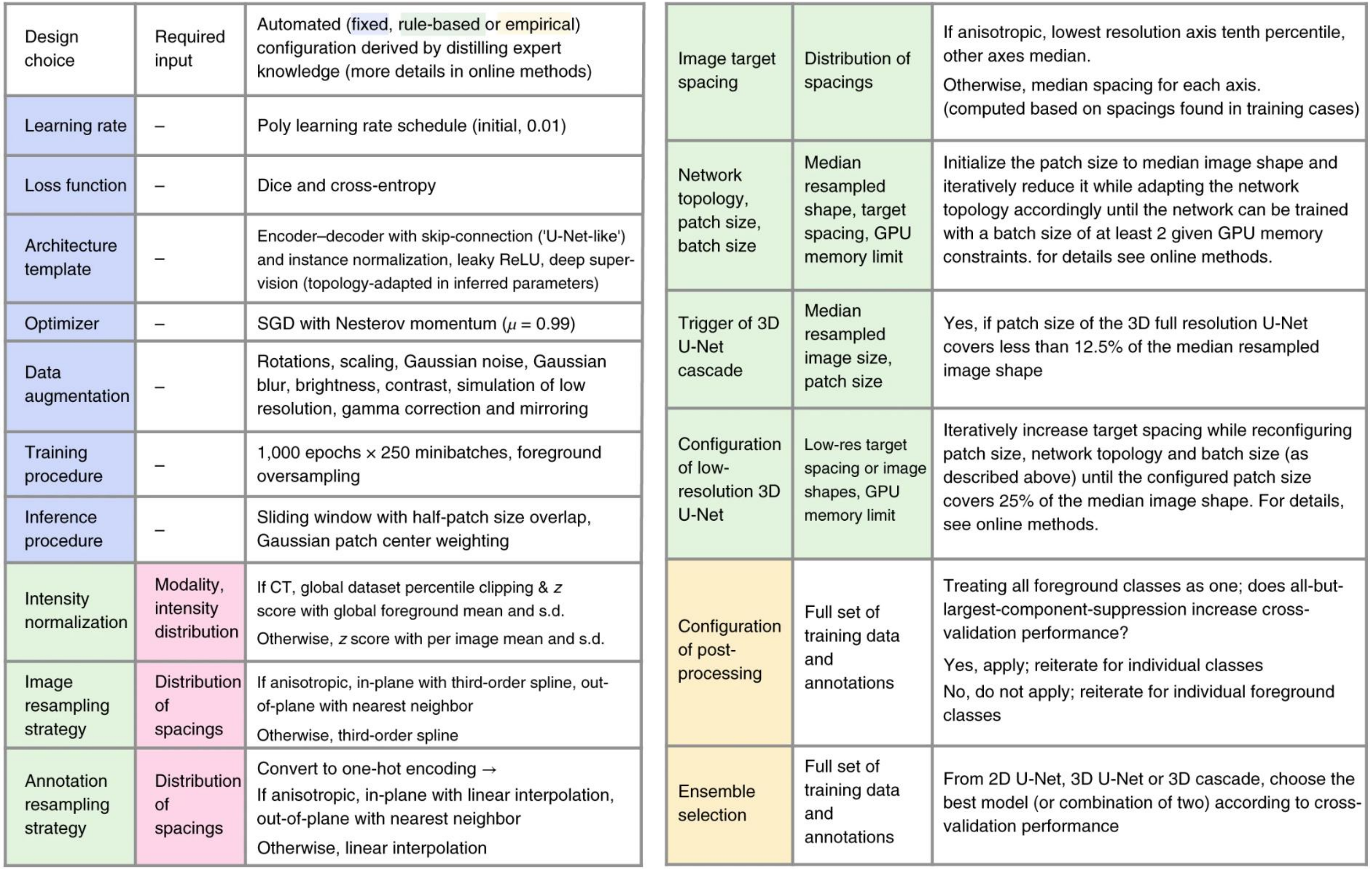


Edema (D1, MRI)
0.6
0.4
0.2
(DC)
Rank
3 of 25
Non-enhancing tumor (D1, MRI)
0.4
0.2
(DC)
Rank
3 of 25
Enhancing. tumor (D1, MRI)
0.6
0.4
0.2
(DC)
Rank
2 of 25
Left atrium (D2, MRI)
0.9
0.8
0.7
(DC)
Rank
1 of 25
Liver (D3, CT)
0.9
0.8
0.7
(DC)
Rank
1 of 25
Liver tumor (D3, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
2 of 25
Anterior hippocampus (D4, MRI)
0.90
0.85
0.80
0.75
(DC)
Rank
1 of 25
Posterior hippocampus (D4, MRI)
0.85
0.80
0.75
(DC)
Rank
1 of 25
Prostate, peripheral zone (D5, MRI)
0.7
0.6
0.5
0.4
(DC)
Rank
1 of 25
Prostate, transition zone (D5, MRI)
0.9
0.8
0.7
(DC)
Rank
1 of 25
Lung nodules (D6, CT)
0.6
0.4
0.2
(DC)
Rank
1 of 25
Pancreas (D7, CT)
0.8
0.6
0.4
(DC)
Rank
1 of 25
Pancreatic cancer (D7, CT)
0.4
0.2
(DC)
Rank
1 of 25
Liver vessel (D8, CT)
0.6
0.4
0.2
(DC)
Rank
1 of 25
Liver tumor (D8, CT)
0.6
0.4
0.2
(DC)
Rank
1 of 25
Spleen (D9, CT)
0.95
0.90
0.85
(DC)
Rank
1 of 25
Colon cancer (D10, CT)
0.6
0.4
0.2
(DC)
Rank
1 of 25
Left adrenal gland (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
2 of 38
Right adrenal gland (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
1 of 38
Aorta (D11, CT)
0.75
0.50
0.25
(DC)
Rank
1 of 40
Esophagus (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
1 of 39
Gallbladder (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
2 of 42
Inferior vena cava (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
1 of 40
Left kidney (D11, CT)
1.00
0.75
0.50
0.25
(DC)
Rank
1 of 45
Right kidney (D11, CT)
0.75
0.50
0.25
(DC)
Rank
1 of 42
Liver (D11, CT)
1.00
0.75
0.50
0.25
(DC)
Rank
1 of 46
Pancreas (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
1 of 42
Portal and splenic vein (D11, CT)
0.8
0.6
0.4
0.2
(DC)
Rank
2 of 41

nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation



nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation



Design choice	Required input	Automated (fixed, rule-based or empirical) configuration derived by distilling expert knowledge (more details in online methods)
Learning rate	–	Poly learning rate schedule (initial, 0.01)
Loss function	–	Dice and cross-entropy
Architecture template	–	Encoder–decoder with skip-connection ('U-Net-like') and instance normalization, leaky ReLU, deep super-vision (topology-adapted in inferred parameters)
Optimizer	–	SGD with Nesterov momentum (μ = 0.99)
Data augmentation	–	Rotations, scaling, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, gamma correction and mirroring
Training procedure	–	1,000 epochs × 250 minibatches, foreground oversampling
Inference procedure	–	Sliding window with half-patch size overlap, Gaussian patch center weighting
Intensity normalization	Modality, intensity distribution	If CT, global dataset percentile clipping & z score with global foreground mean and s.d. Otherwise, z score with per image mean and s.d.
Image resampling strategy	Distribution of spacings	If anisotropic, in-plane with third-order spline, out-of-plane with nearest neighbor Otherwise, third-order spline
Annotation resampling strategy	Distribution of spacings	Convert to one-hot encoding → If anisotropic, in-plane with linear interpolation, out-of-plane with nearest neighbor Otherwise, linear interpolation
Image target spacing	Distribution of spacings	If anisotropic, lowest resolution axis tenth percentile, other axes median. Otherwise, median spacing for each axis. (computed based on spacings found in training cases)
Network topology, patch size, batch size	Median resampled shape, target spacing, GPU memory limit	Initialize the patch size to median image shape and iteratively reduce it while adapting the network topology accordingly until the network can be trained with a batch size of at least 2 given GPU memory constraints. for details see online methods.
Trigger of 3D U-Net cascade	Median resampled image size, patch size	Yes, if patch size of the 3D full resolution U-Net covers less than 12.5% of the median resampled image shape
Configuration of low-resolution 3D U-Net	Low-res target spacing or image shapes, GPU memory limit	Iteratively increase target spacing while reconfiguring patch size, network topology and batch size (as described above) until the configured patch size covers 25% of the median image shape. For details, see online methods.
Configuration of post-processing	Full set of training data and annotations	Treating all foreground classes as one; does all-but-largest-component-suppression increase cross-validation performance? Yes, apply; reiterate for individual classes No, do not apply; reiterate for individual foreground classes
Ensemble selection	Full set of training data and annotations	From 2D U-Net, 3D U-Net or 3D cascade, choose the best model (or combination of two) according to cross-validation performance

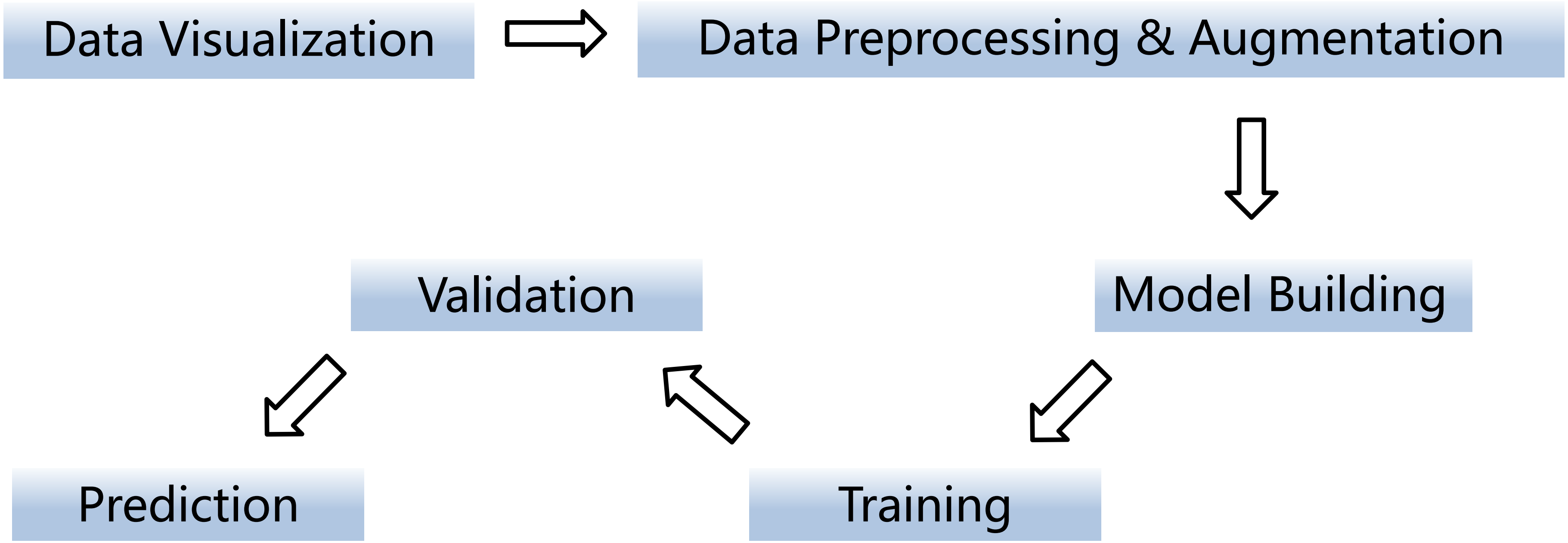
PART 02

nnU-Netv2 Implementation

Objective

Medical image segmentation plays a crucial role in clinical diagnosis and treatment planning. **Deep learning-based segmentation** models, particularly the 3D **U-Net** architecture, have demonstrated remarkable performance in accurately delineating anatomical structures from MRI scans. This work presents a segmentation pipeline built upon **nnUNetv2**, incorporating advanced preprocessing, data augmentation, and a customized training framework to enhance model robustness and accuracy.

Brain-Tumor Segmentation Process



Data Visualization

Data Preprocessing & Augmentation

Model Building

Training

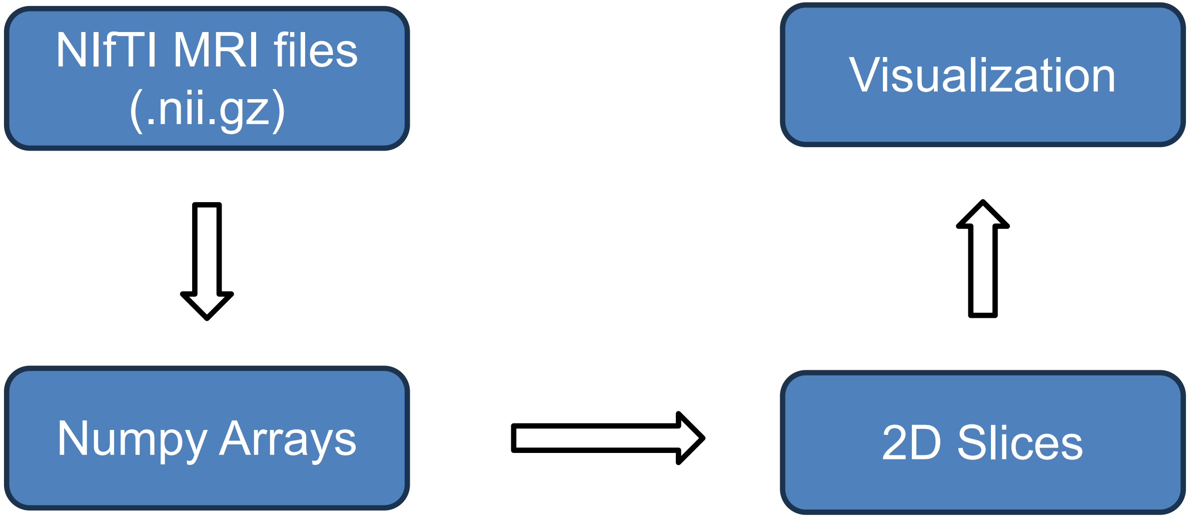
Validation

Prediction

PART 03

MRI Volume Visualization

Data Visualization



NIfTI MRI files (.nii.gz)

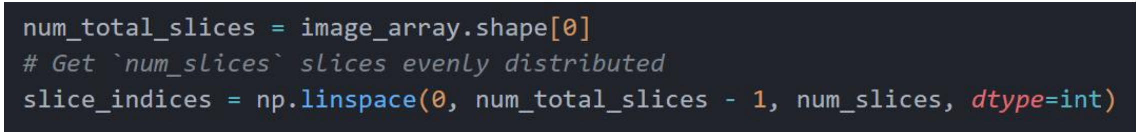
Numpy Arrays

2D Slices

Visualization

[Z, Y, X]

Z: number of axial slices in the MRI scan



```
num_total_slices = image_array.shape[0]
# Get `num_slices` slices evenly distributed
slice_indices = np.linspace(0, num_total_slices - 1, num_slices, dtype=int)
```

Prase the Data

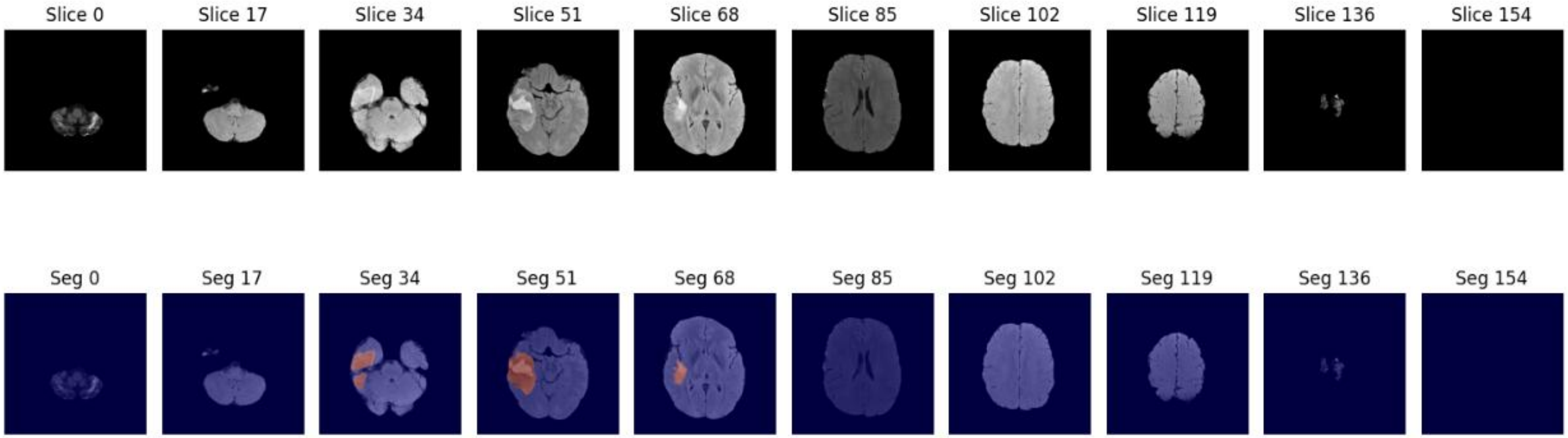
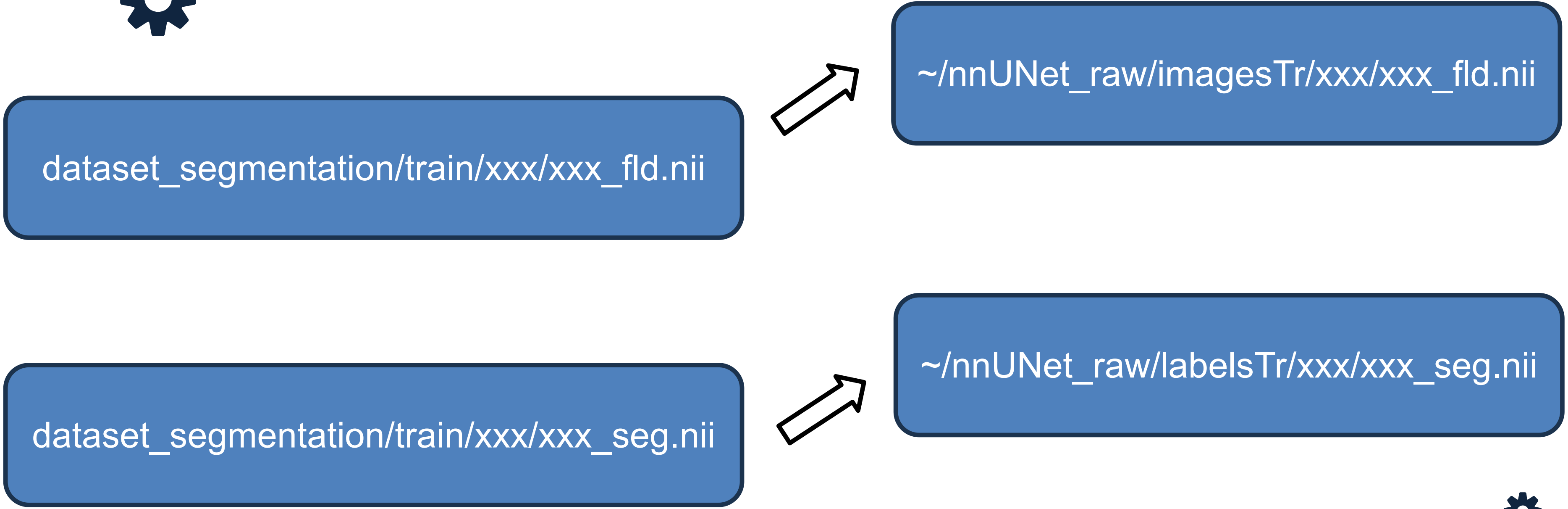


Figure 1: Example Visualization

PART 04

Data Preprocessing

Track 2.1 Restructuring Dataset



dataset_segmentation/train/xxx/xxx_fld.nii

~/nnUNet_raw/imagesTr/xxx/xxx_fld.nii

dataset_segmentation/train/xxx/xxx_seg.nii

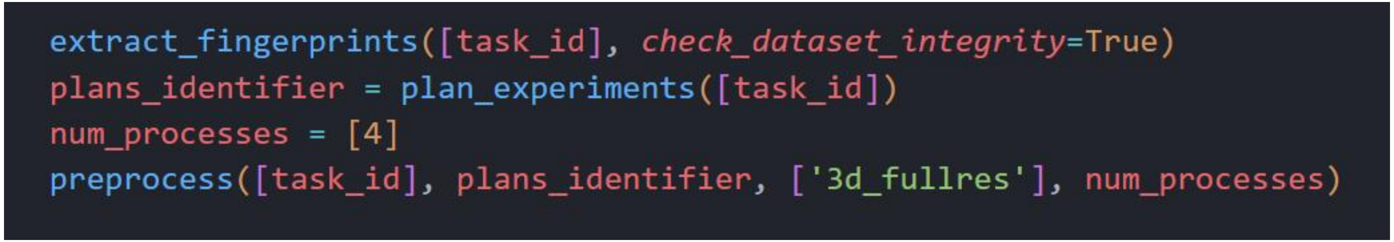
~/nnUNet_raw/labelsTr/xxx/xxx_seg.nii

Track 2.2 Extracting Data Fingerprint

Cropping Non-Zero Regions

Non-zero mask is created for each image and segmentation pair. This mask ensures that only the relevant part of the image is used for training.

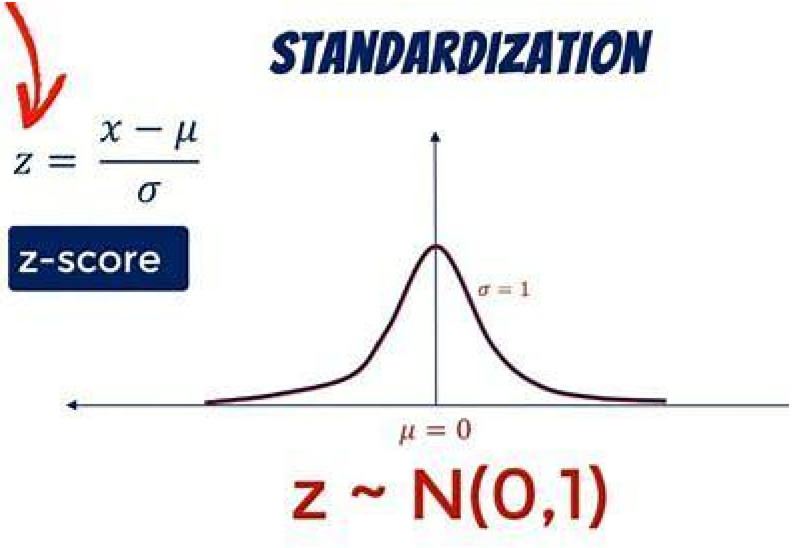
Saving the Fingerprint



```
extract_fingerprints([task_id], check_dataset_integrity=True)
plans_identifier = plan_experiments([task_id])
num_processes = [4]
preprocess([task_id], plans_identifier, ['3d_fullres'], num_processes)
```

Track 2.3 Preprocessing Individual Cases

Normalization



Resampling

The original spacing (voxel size) of the images is adjusted to a targetspacing.

PART 05

Data Augmentation

Augmentation Techniques

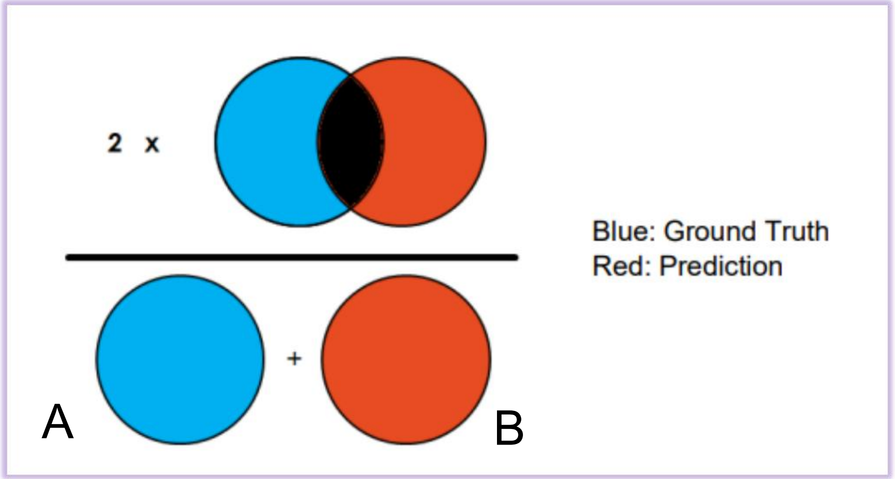
Patch Selection	Fixed-size patches
Spatial Transformations	Rotation, Scaling, Elastic Deformation
Noise & Blur	
Brightness and Contrast	
Low Resolution & Gamma Transform	• Downsampling • Adjust intensity distribution
Mirroring and Masking	• Applied specified axes • Normalizes only within regions of interest

PART 06

Performance Metric

4.1 Dice Similarity Coefficient

$$DSC = \frac{2 \cdot |A \cap B| + \text{smoothing}}{|A| + |B| + \text{smoothing}}$$



4.2 Cross-Entropy Loss

$$H(p,q) = -\sum_{x\in\mathcal{X}} p(x)\,\log q(x). \quad \textbf{(Eq.1)}$$

Supports One-Hot Encoding	Works with one-hot encoded target labels
Ensures Pixel Classification	Helps the model assign each pixel to the right class

4.3 Combined Loss Function

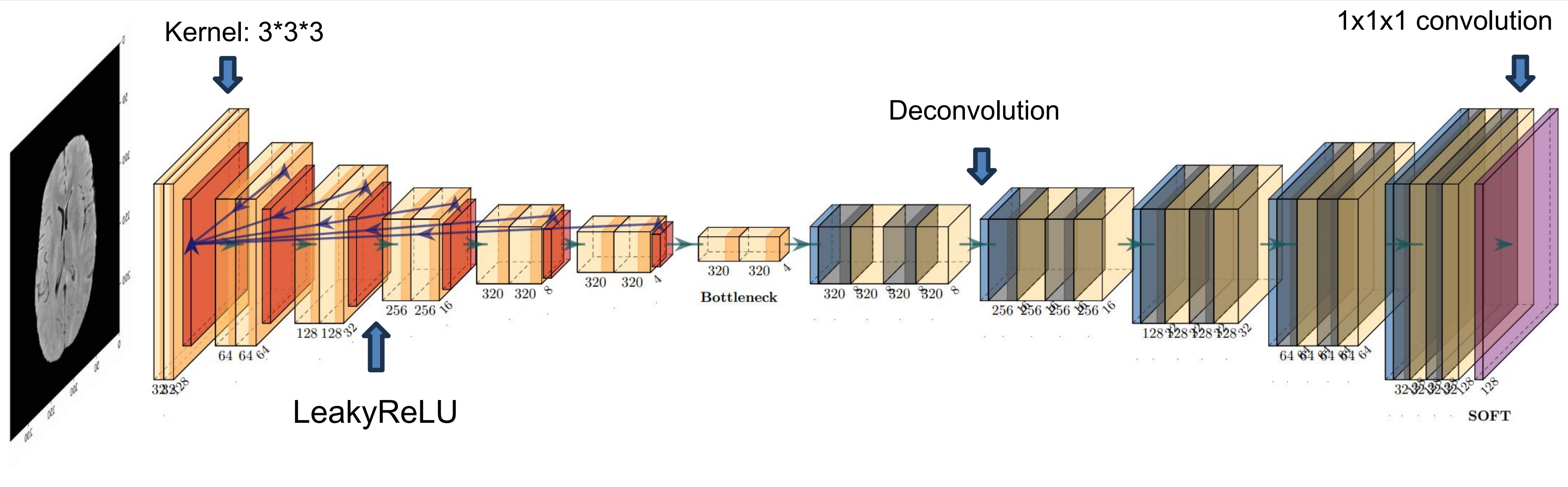
$$\mathcal{L} = w_{\text{dice}} \cdot \mathcal{L}_{\text{Dice}} + w_{\text{CE}} \cdot \mathcal{L}_{\text{CE}}$$

Ensures both accurate **classification** and well-defined **segmentation boundaries**.

PART 07

Network Architecture

PlainConvUNet

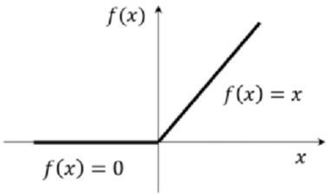


Kernel: 3*3*3

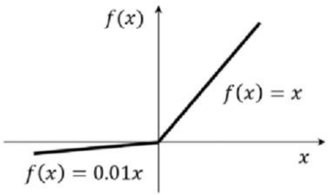
LeakyReLU

Deconvolution

1x1x1 convolution



ReLU activation function



LeakyReLU activation function

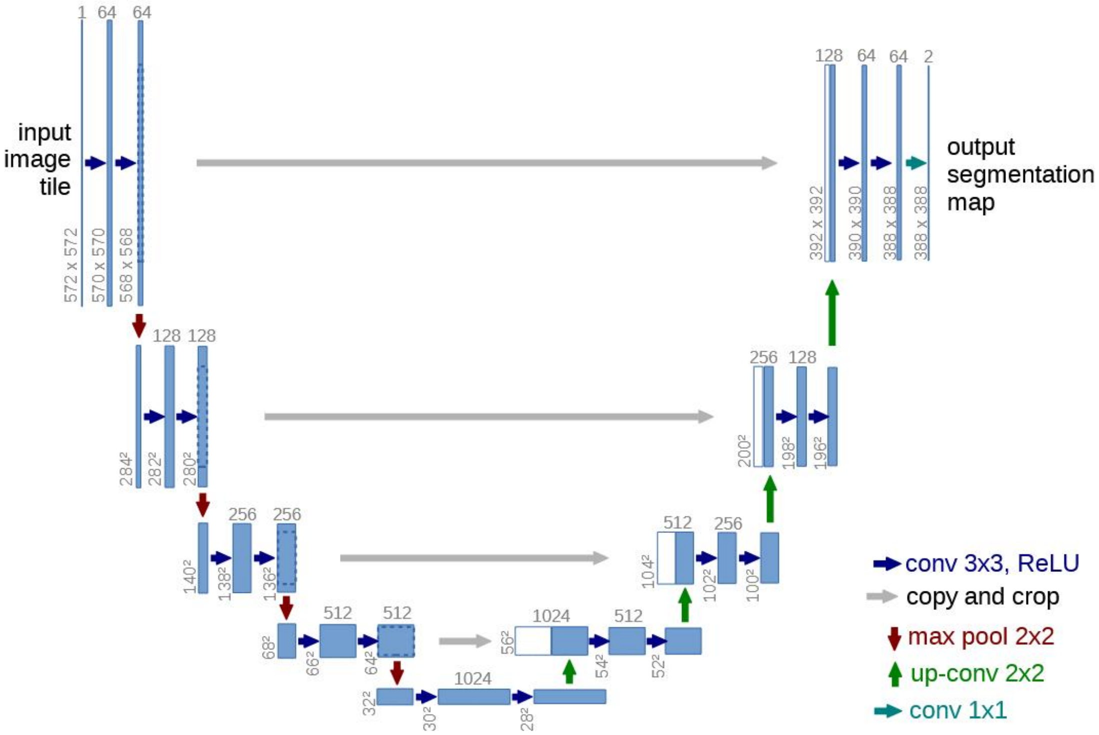
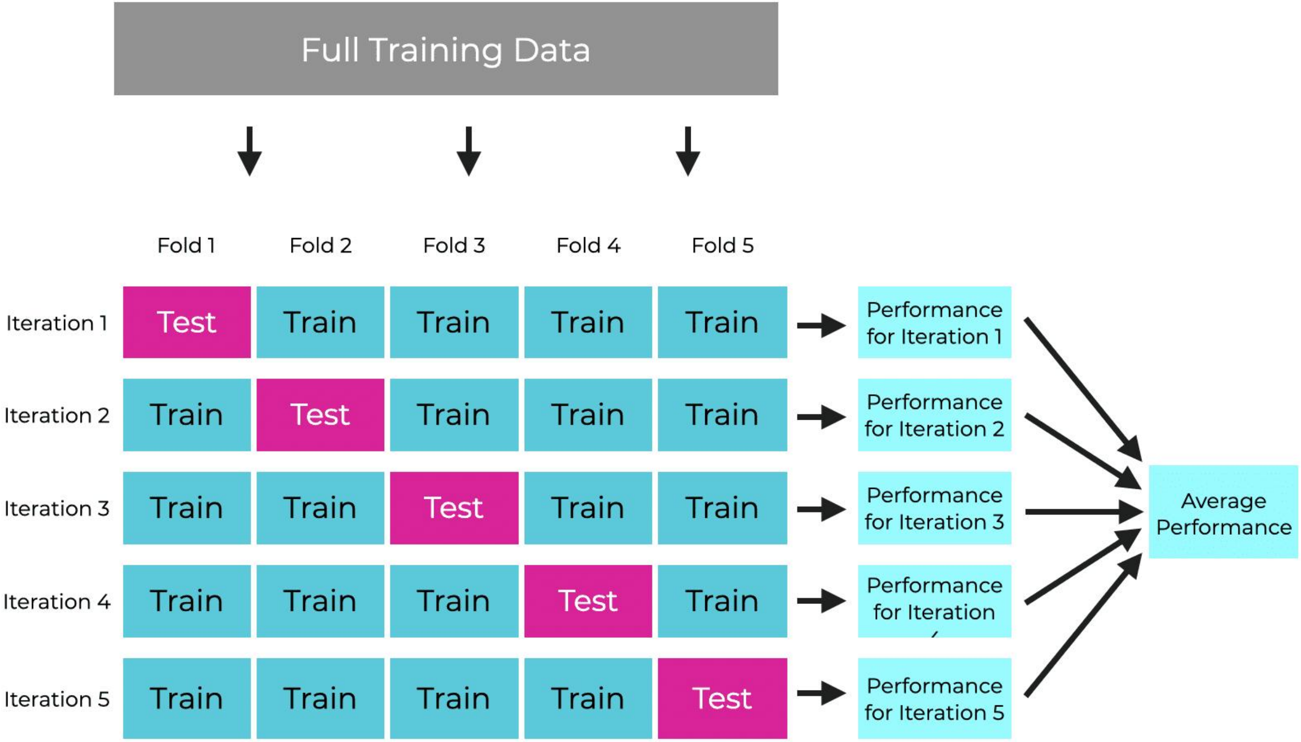


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

PART 08

Trainer

5-Fold Cross-Validation

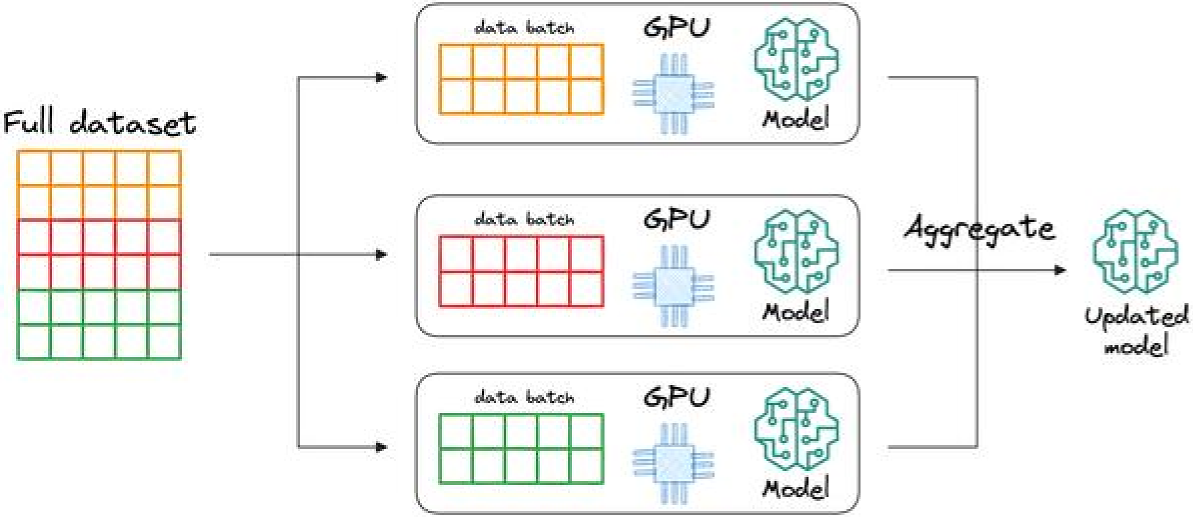


Full Training Data

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Iteration 1	Test	Train	Train	Train	Train	Performance for Iteration 1
Iteration 2	Train	Test	Train	Train	Train	Performance for Iteration 2
Iteration 3	Train	Train	Test	Train	Train	Performance for Iteration 3
Iteration 4	Train	Train	Train	Test	Train	Performance for Iteration
Iteration 5	Train	Train	Train	Train	Test	Performance for Iteration 5

Average Performance

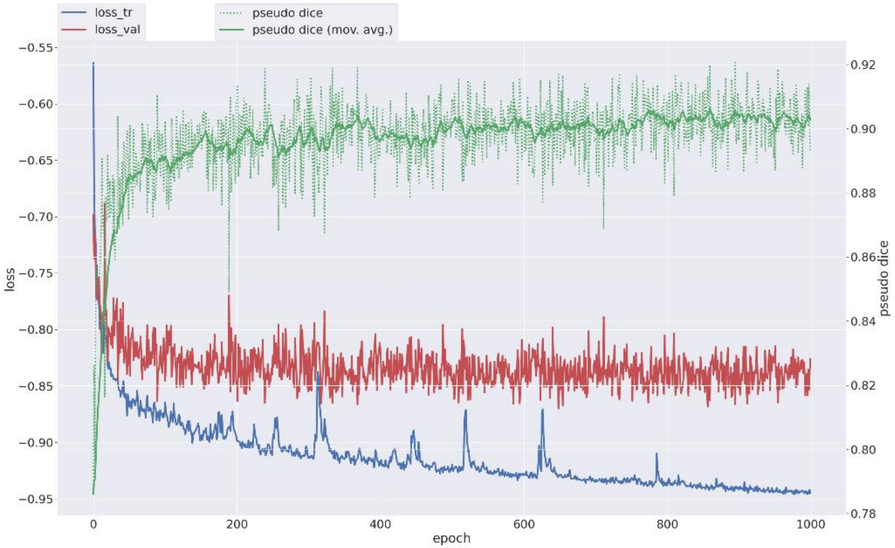
Multi-GPU Training

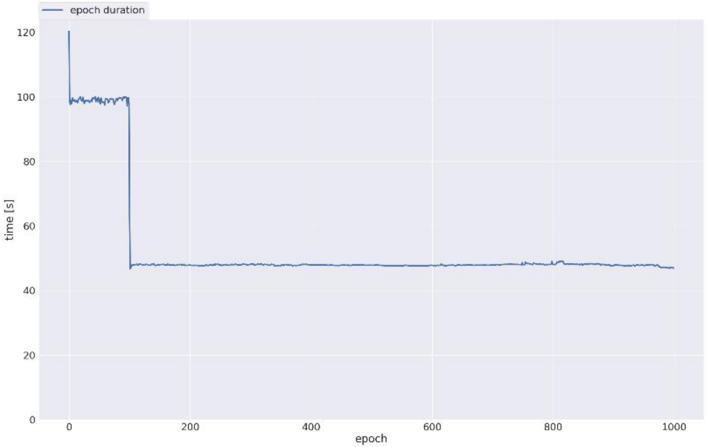


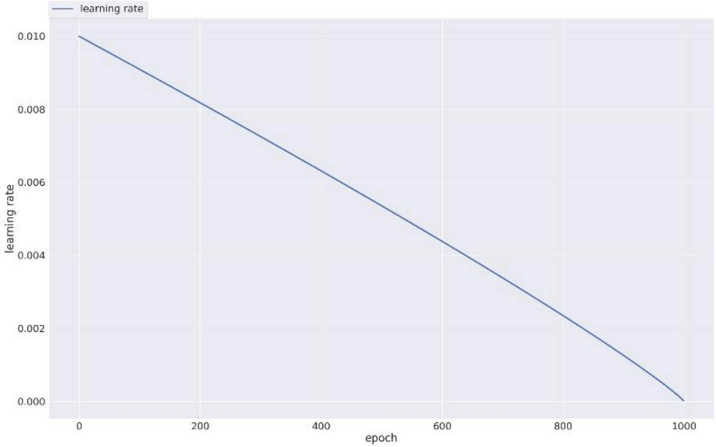
PART 09

Results

Training Process

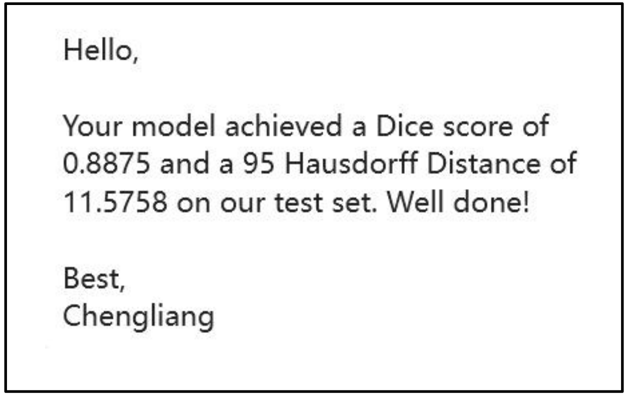






Validation Performance

- **Fold 0:** Mean Validation Dice: 0.9173
- **Fold 1:** Mean Validation Dice: 0.8910
- **Fold 2:** Mean Validation Dice: 0.8888
- **Fold 3:** Mean Validation Dice: 0.8911
- **Fold 4:** Mean Validation Dice: 0.9103
- **Average Mean Dice Score:** 0.8997



Hello,

Your model achieved a Dice score of 0.8875 and a 95 Hausdorff Distance of 11.5758 on our test set. Well done!

Best,
Chengliang

Imperial Data Science and AI Winter School 2025

Word Representation in **Biomedical Domain**

Group 3
Li Zhuoran Chen Xinjia Xu Zhesheng

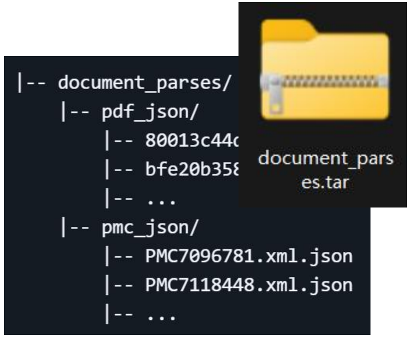
Objective

The aim of this project is to **analyze the text** content and **construct word representations** suitable for the large scale document data in the **biomedical** field, and finally explore and analyze the **application** of word vectors.

Technological Process

Imperial DS&AI Winter School

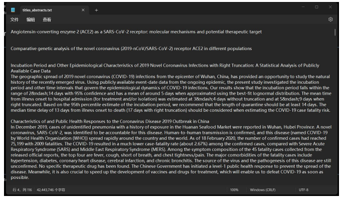
Dataset



```
|-- document_parses/
    |-- pdf_json/
        |-- 80013c44c
        |-- bfe20b358
        |-- ...
    |-- pmc_json/
        |-- PMC7096781.xml.json
        |-- PMC7118448.xml.json
        |-- ...
```

https://www.semanticscholar.org/cord19/

Core content



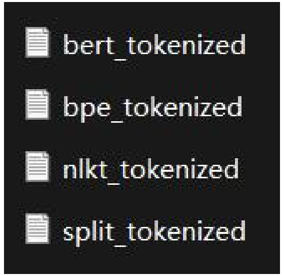
Angiotensin-converting enzyme 2 (ACE2) as a SARS-CoV-2 receptor: molecular mechanisms and potential therapeutic target

Comparative genetic analysis of the novel coronavirus (2019-nCoV/SARS-CoV-2) receptor ACE2 in different populations

Incubation Period and Other Epidemiological Characteristics of 2019 Novel Coronavirus Infections with Right Truncation: A Statistical Analysis of Publicly Available Case Data
The geographic spread of 2019 novel coronavirus (COVID-19) infections from the epicenter of Wuhan, China, has provided an opportunity to study the natural history of the recently emerged virus. Using publicly available event-date data from the ongoing epidemic, the present study investigated the incubation period and other time intervals that govern the epidemiological dynamics of COVID-19 infections. Our results show that the incubation period falls within the range of 2&ndash;14 days with 95% confidence and has a mean of around 5 days when approximated using the best-fit lognormal distribution. The mean time from illness onset to hospital admission (for treatment and/or isolation) was estimated at 3&ndash;4 days without truncation and at 5&ndash;9 days when right truncated. Based on the 95th percentile estimate of the incubation period, we recommend that the length of quarantine should be at least 14 days. The median time delay of 13 days from illness onset to death (17 days with right truncation) should be considered when estimating the COVID-19 case fatality risk.

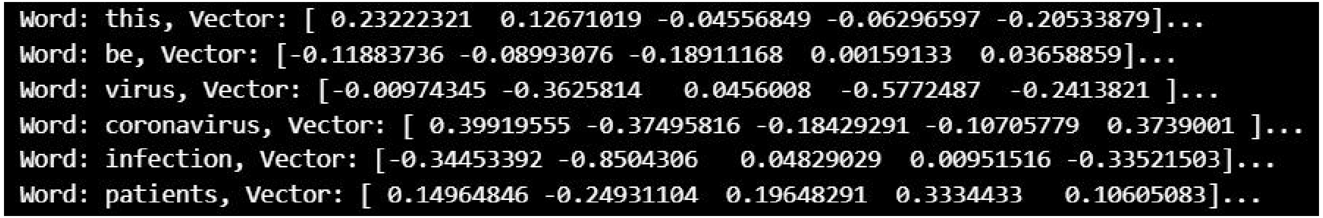
Characteristics of and Public Health Responses to the Coronavirus Disease 2019 Outbreak in China
In December 2019, cases of unidentified pneumonia with a history of exposure in the Huanan Seafood Market were reported in Wuhan, Hubei Province. A novel coronavirus, SARS-CoV-2, was identified to be accountable for this disease. Human-to-human transmission is confirmed, and this disease (named COVID-19 by World Health Organization (WHO)) spread rapidly around the country and the world. As of 18 February 2020, the number of confirmed cases had reached 75,199 with 2009 fatalities. The COVID-19 resulted in a much lower case-fatality rate (about 2.67%) among the confirmed cases, compared with Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS). Among the symptom composition of the 45 fatality cases collected from the released official reports, the top four are fever, cough, short of breath, and chest tightness/pain. The major comorbidities of the fatality cases include hypertension, diabetes, coronary heart disease, cerebral infarction, and chronic bronchitis. The source of the virus and the pathogenesis of this disease are still unconfirmed. No specific therapeutic drug has been found. The Chinese Government has initiated a level-1 public health response to prevent the spread of the disease. Meanwhile, it is also crucial to speed up the development of vaccines and drugs for treatment, which will enable us to defeat COVID-19 as soon as possible.

Tokenizer



- bert_tokenized
- bpe_tokenized
- nlkt_tokenized
- split_tokenized

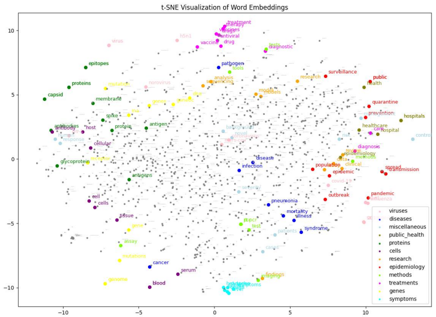
Modeling



```
Word: this, Vector: [ 0.23222321  0.12671019 -0.04556849 -0.06296597 -0.20533879]...
Word: be, Vector: [-0.11883736 -0.08993076 -0.18911168  0.00159133  0.03658859]...
Word: virus, Vector: [-0.00974345 -0.3625814   0.0456008  -0.5772487  -0.2413821 ]...
Word: coronavirus, Vector: [ 0.39919555 -0.37495816 -0.18429291 -0.10705779  0.3739001 ]...
Word: infection, Vector: [-0.34453392 -0.8504306   0.04829029  0.00951516 -0.33521503]...
Word: patients, Vector: [ 0.14964846 -0.24931104  0.19648291  0.3334433   0.10605083]...
```

Representations

Visualisation & Application



t-SNE Visualization of Word Embeddings

PART 01

Prase the Data

Prase the Data

Overview of Dataset

metadata.csv

|--**title**

|--**abstract**

|--authors

|--doi

……

36.3G

Selection of Content

- **Title**
- **Abstract**

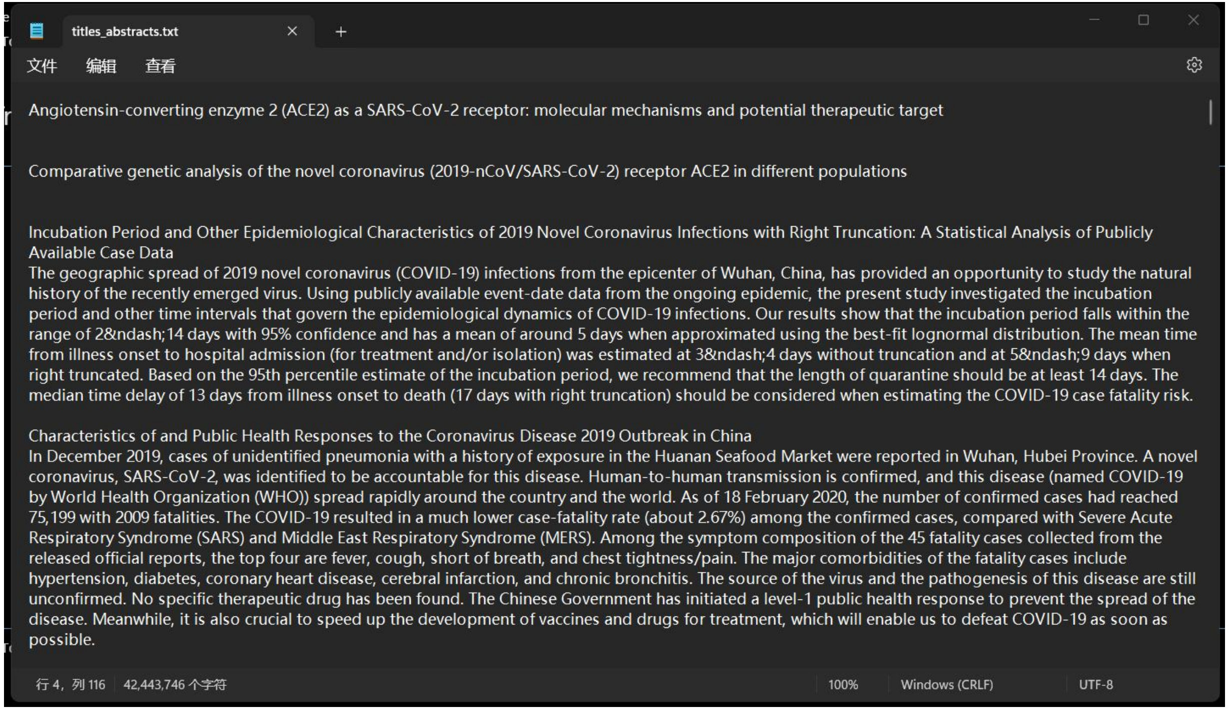
*If missing,
replaced with a
‘Space’

40.6M

Prase the Data

Imperial DS&AI
Winter School

titles_abstracts.txt



Angiotensin-converting enzyme 2 (ACE2) as a SARS-CoV-2 receptor: molecular mechanisms and potential therapeutic target

Comparative genetic analysis of the novel coronavirus (2019-nCoV/SARS-CoV-2) receptor ACE2 in different populations

Incubation Period and Other Epidemiological Characteristics of 2019 Novel Coronavirus Infections with Right Truncation: A Statistical Analysis of Publicly Available Case Data
The geographic spread of 2019 novel coronavirus (COVID-19) infections from the epicenter of Wuhan, China, has provided an opportunity to study the natural history of the recently emerged virus. Using publicly available event-date data from the ongoing epidemic, the present study investigated the incubation period and other time intervals that govern the epidemiological dynamics of COVID-19 infections. Our results show that the incubation period falls within the range of 2&ndash;14 days with 95% confidence and has a mean of around 5 days when approximated using the best-fit lognormal distribution. The mean time from illness onset to hospital admission (for treatment and/or isolation) was estimated at 3&ndash;4 days without truncation and at 5&ndash;9 days when right truncated. Based on the 95th percentile estimate of the incubation period, we recommend that the length of quarantine should be at least 14 days. The median time delay of 13 days from illness onset to death (17 days with right truncation) should be considered when estimating the COVID-19 case fatality risk.

Characteristics of and Public Health Responses to the Coronavirus Disease 2019 Outbreak in China
In December 2019, cases of unidentified pneumonia with a history of exposure in the Huanan Seafood Market were reported in Wuhan, Hubei Province. A novel coronavirus, SARS-CoV-2, was identified to be accountable for this disease. Human-to-human transmission is confirmed, and this disease (named COVID-19 by World Health Organization (WHO)) spread rapidly around the country and the world. As of 18 February 2020, the number of confirmed cases had reached 75,199 with 2009 fatalities. The COVID-19 resulted in a much lower case-fatality rate (about 2.67%) among the confirmed cases, compared with Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS). Among the symptom composition of the 45 fatality cases collected from the released official reports, the top four are fever, cough, short of breath, and chest tightness/pain. The major comorbidities of the fatality cases include hypertension, diabetes, coronary heart disease, cerebral infarction, and chronic bronchitis. The source of the virus and the pathogenesis of this disease are still unconfirmed. No specific therapeutic drug has been found. The Chinese Government has initiated a level-1 public health response to prevent the spread of the disease. Meanwhile, it is also crucial to speed up the development of vaccines and drugs for treatment, which will enable us to defeat COVID-19 as soon as possible.

PART 02

Tokenization

Track 2.1 Use split() with regex

```
import re
```

```
Input: content of file
Initialize empty list words
for each line in content do
    Split line using regex \W+
    Append tokens to words list
end for
Remove empty strings from words
Write result to file
Output: result of tokenization
```

Table 2.1: Weighted Random Sample of 20 Words

source	for
acetate	number
rate	enzymes
peptide	fruit
MERS	with
in	enzymes
of	respiratory
disease	virus
investigation	that
activation	the

Track 2.2 Use NLTK tokenizer

```
import nltk
```

```
Input: content of file
Split content into sentences using
        sent_tokenize()
Initialize empty list words
for each sentence in sentences do
    Tokenize sentence into words
        using word_tokenize()
    Remove punctuations from sentence
    Convert all words to lowercase
    Append processed words to words
end for
Write result to file
Output: result of tokenization
```

Table 2.2: Top 20 Representative High-Frequency Words of Tokenized Result of NLTK Tokenizer

covid-19	sars-cov-2
2019-n-ncov	infection
pneumonia	transmission
symptoms	outbreak
wuhan	china
case	virus
fever	cough
diagnosis	treatment
epidemic	prevention
control	clinical

Track 2.3 Use Byte-Pair Encoding (BPE)

```
from transformers
import AutoTokenizer
```

```
Input: content of file
Load pre-trained BERT tokenizer
Set max_length to 512
Initialize empty list result
for each chunk of text with length max_length do
    Tokenize chunk using BERT tokenizer
    Append tokenized chunk to result
end for
Write result to file
Output: result of tokenization
```

Table 2.3: Top 20 Representative High-Frequency Words of Tokenized Result of BERT Tokenizer

COVID-19	SARS-CoV-2
2019-nCoV	virus
infection	pneumonia
symptoms	transmission
patients	diagnosis
treatment	research
outbreak	cases
china	wuhan
genome	epidemiology
clinical	prevention

Track 2.4 Build custom BPE

```
from tokenizers import
Tokenizer, models, trainers,
pre_tokenizers
```

```
Input: content of file
Initialize BPE model
Set up the trainer for the model
Training BPE model with content
Save the trained BPE model
Tokenize content using trained
BPE model (as Track 2.3)
Write result to file
Output: result of tokenization
```

Table 2.4: Top 20 Representative High-Frequency Words of Tokenized Result of Trained BPE Model Tokenizer

corona	virus
infection	COVID
pneumonia	epidemic
transmission	cases
patients	symptoms
treatment	control
quarantine	diagnosis
outbreak	SARS
MERS	vaccine
mortality	epidemiology

Pros and Cons

Track	Efficiency	Accuracy	Domain Suitability	Ease of Use	Adaptability
Use split() with regex	High	Low	Low	Very High	None
Use NLTK tokenizer	Moderate	Moderate	Moderate	Moderate	Moderate
Use Byte-Pair Encoding (BPE)	High	Moderate	Moderate	Low	Low
Build new Byte-Pair Encoding	Moderate	Very High	Very High	Low	Very High

PART 03

Build Word Representations

Track 3.1 Use N-gram Language Modeling

N-gram language modeling represents words using **statistical** co-occurrence.

n=7

the	main	symptoms	of	COVID-19	are	?
ω_{t-6}	ω_{t-5}	ω_{t-4}	ω_{t-3}	ω_{t-2}	ω_{t-1}	ω_t

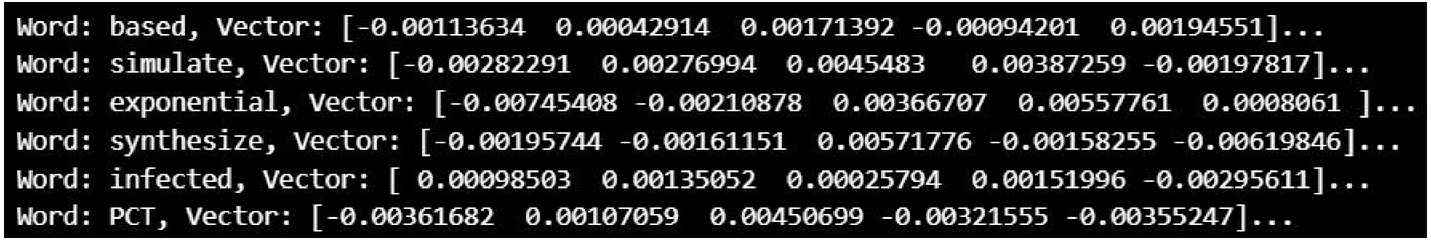
$$\boxed{p(\omega_t|\omega_{t-n+1},\dots,\omega_{t-1}) = \frac{C(\omega_{t-n+1},\dots,\omega_{t-1},\omega_t)}{C(\omega_{t-n+1},\dots,\omega_{t-1})}}$$

*C counts the number of occurrences of the sequence

Embeddings are learned by constructing a co-occurrence matrix $M_{ij} = p(\omega_j|\omega_i)$. Embeddings are obtained by SVD decomposition of M: $M = USV^T$, where U is word embeddings and V is context embeddings.

Track 3.1 Use N-gram Language Modeling

```
Input: result of tokenization
Extract vocabulary from result
Count the number of occurrences of different sequenses
Initialize empty matrix M
for each word-pair (wi,wj) in vocabulary do
    Caculate Mij = count of (wi,wj) devided by count of wi
Obtain embeddings U and V by SVD decomposition of M
Write representations to file
Output: word representations
```

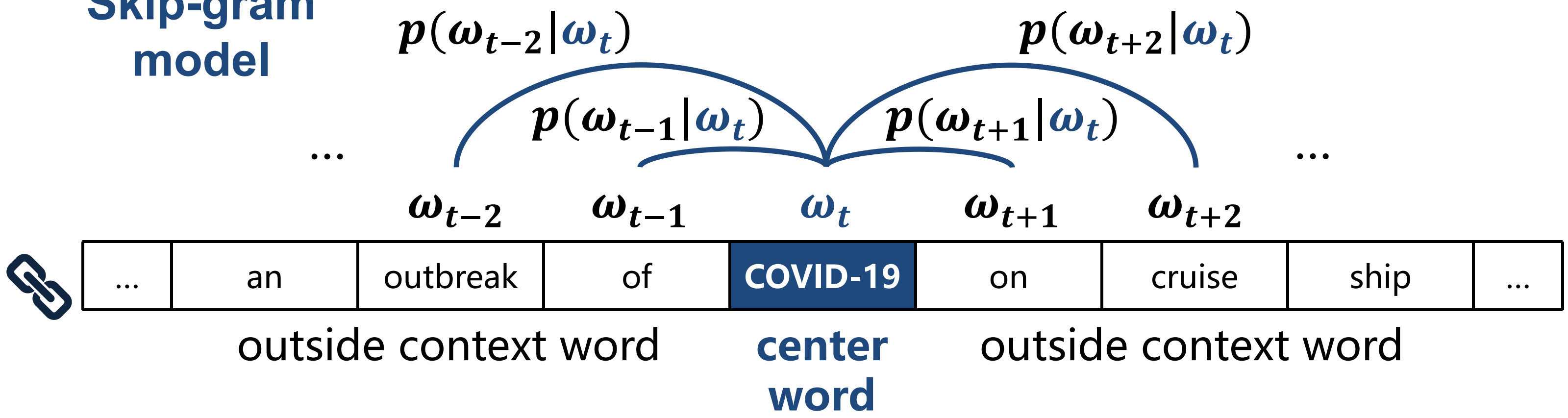


```
Word: based, Vector: [-0.00113634  0.00042914  0.00171392 -0.00094201  0.00194551]...
Word: simulate, Vector: [-0.00282291  0.00276994  0.0045483   0.00387259 -0.00197817]...
Word: exponential, Vector: [-0.00745408 -0.00210878  0.00366707  0.00557761  0.0008061 ]...
Word: synthesize, Vector: [-0.00195744 -0.00161151  0.00571776 -0.00158255 -0.00619846]...
Word: infected, Vector: [ 0.00098503  0.00135052  0.00025794  0.00151996 -0.00295611]...
Word: PCT, Vector: [-0.00361682  0.00107059  0.00450699 -0.00321555 -0.00355247]...
```

Track 3.2 Use Skip-gram with Negative Sampling

Skip-gram with Negative Sampling (SGNS) learns word embeddings by training a **neural network** to predict **context words** given a **target word**, while also distinguishing real context words from randomly sampled **negative words**.

Skip-gram model



$p(\omega_{t-2}|\omega_t)$ $p(\omega_{t+2}|\omega_t)$

$p(\omega_{t-1}|\omega_t)$ $p(\omega_{t+1}|\omega_t)$

… ω_{t-2} ω_{t-1} ω_t ω_{t+1} ω_{t+2} …

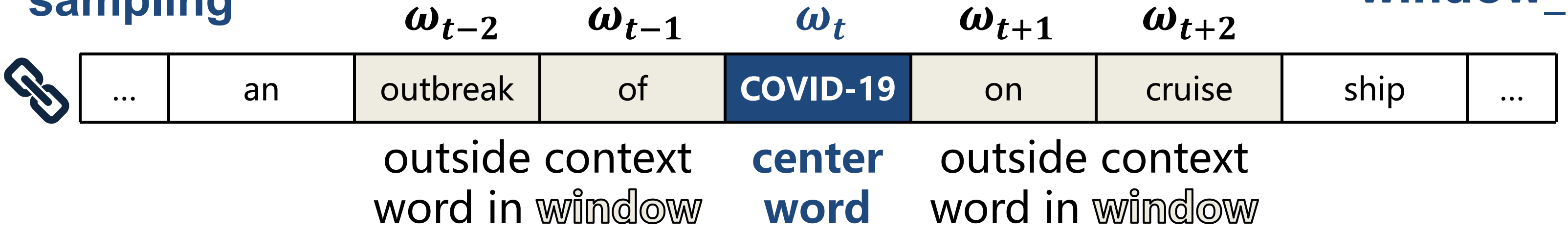
…	an	outbreak	of	**COVID-19**	on	cruise	ship	…

outside context word **center word** outside context word

Track 3.2 Use Skip-gram with Negative Sampling

Nagetive sampling

window_size=5



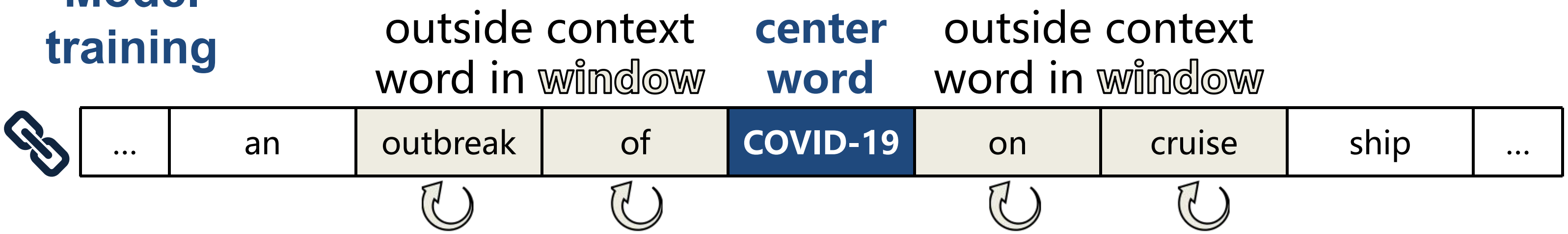
- We use $p(\omega_c|\omega_t) = \sigma(v_c^T v_t)$ to calculate $p(\omega_c|\omega_t)$ with v_c(the vector of context word ω_c), v_t(the vector of target word ω_t) and function $\sigma(x) = \frac{1}{1+e^{-x}}$

- Outside context words in window are selected as **positive samples**, while k **negative samples** ω_n are drawn from a noise distribution.

- To maximize $p(\omega_c|\omega_t)$ while minimize $p(\omega_n|\omega_t)$, the loss function simplifies to: $L = -\log\sigma(v_c^T v_t) - \sum_{n=1}^{k} \log\sigma(-v_n^T v_t)$

Track 3.2 Use Skip-gram with Negative Sampling

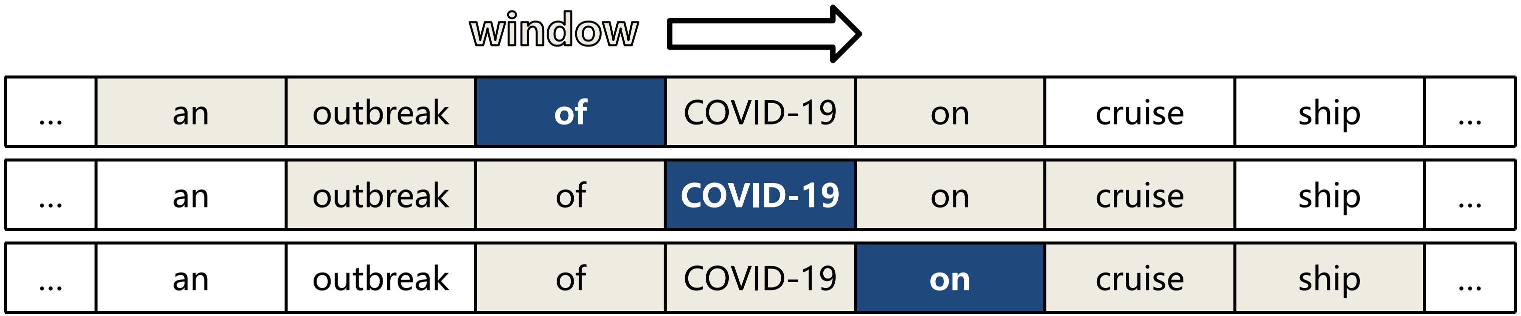
Model training



outside context word in window | center word | outside context word in window

...	an	outbreak	of	**COVID-19**	on	cruise	ship	...

For each context word, we update the vectors once.
Then we move the window forward, traversing the entire content.

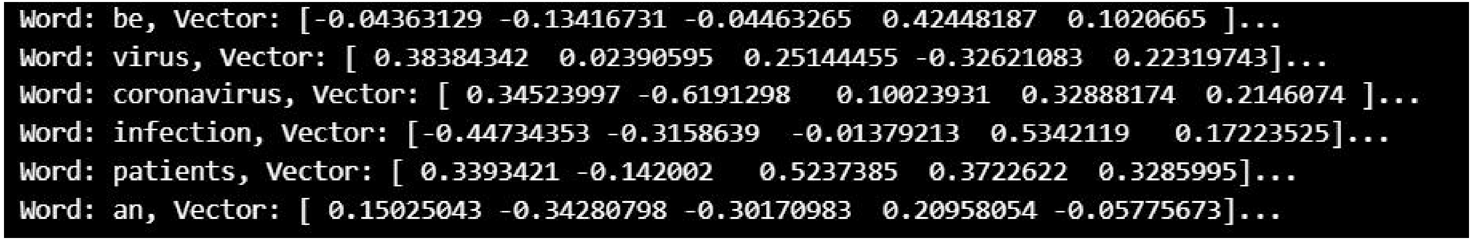


window

...	an	outbreak	**of**	COVID-19	on	cruise	ship	...
...	an	outbreak	of	**COVID-19**	on	cruise	ship	...
...	an	outbreak	of	COVID-19	**on**	cruise	ship	...

Track 3.2 Use Skip-gram with Negative Sampling

```
Input: result of tokenization,window size,k
Initialize vectors randomly
for each sentence in result do
    for each target word in sentence do
        Set the window with the target word as the center
        for each context word in window do
            Randomly select k negative samples
            Caculate the loss function
            Update vectors based on gradient
Write model to file
Output: word representations
```



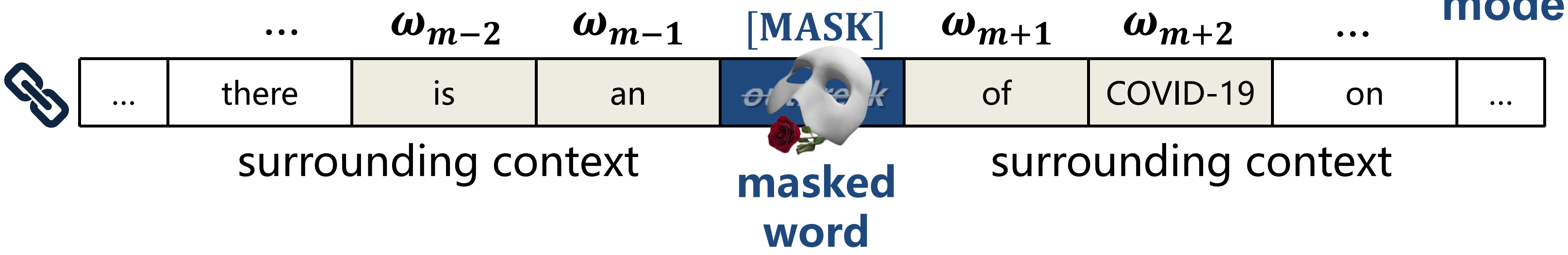
```
Word: be, Vector: [-0.04363129 -0.13416731 -0.04463265  0.42448187  0.1020665 ]...
Word: virus, Vector: [ 0.38384342  0.02390595  0.25144455 -0.32621083  0.22319743]...
Word: coronavirus, Vector: [ 0.34523997 -0.6191298   0.10023931  0.32888174  0.2146074 ]...
Word: infection, Vector: [-0.44734353 -0.3158639  -0.01379213  0.5342119   0.17223525]...
Word: patients, Vector: [ 0.3393421 -0.142002    0.5237385  0.3722622  0.3285995]...
Word: an, Vector: [ 0.15025043 -0.34280798 -0.30170983  0.20958054 -0.05775673]...
```

Track 3.3 Use Contextualised Word Representation by MLM

Masked Language Model (MLM) is a technique used to learn contextualized word representations by training a model to predict randomly **masked words** in a sentence based on their **surrounding context**.

Given an input sequence $\omega_1, \omega_2, \ldots, \omega_t$, some tokens are randomly masked (e.g. ω_m), and the model learns to predict them: $p(\omega_m | \omega_1, \ldots, \omega_{m-1}, [\mathbf{MASK}], \omega_{m+1}, \ldots, \omega_t)$

Masked language model



Track 3.3 Use Contextualised Word Representation by MLM

- A transformer-based model (e.g. BERT) processes the full sequence bidirectionally, generating deep contextualized word embeddings. The loss function is typically the cross-entropy loss over the masked tokens:

$$L =- \sum_m \log \boldsymbol{p}(\boldsymbol{\omega}_m | \boldsymbol{h}_m)$$

where $\boldsymbol{h}_m$ is the hidden representation of the masked token.

- This method enables embeddings to **capture word sense variations and syntactic dependencies** based on the given context.

Track 3.3 Use Contextualised Word Representation by MLM

- In our implementation, we utilized BERT for embedding generation. Given the large number of parameters in BERT and the high computational cost of training, we employed **LoRA fine-tuning** to optimize the training process.

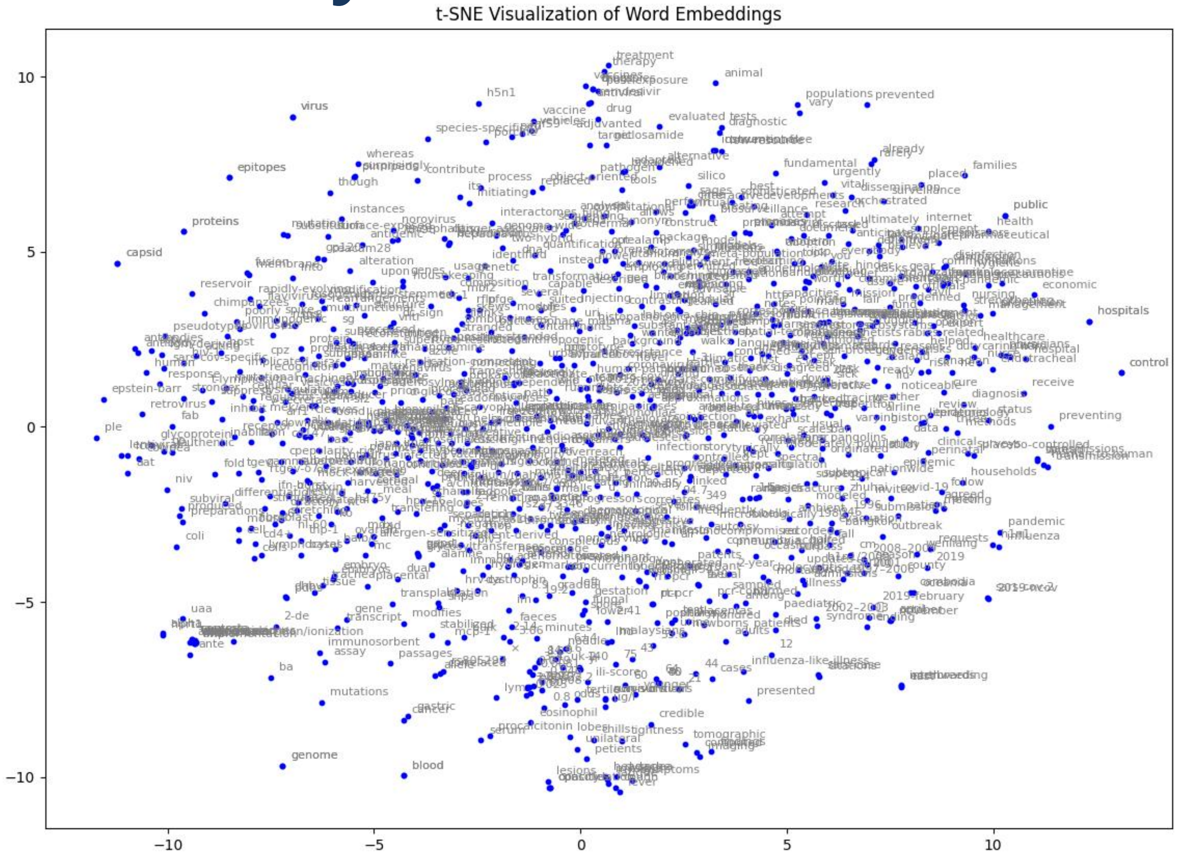
```
from transformers import BertTokenizer, BertForMaskedLM,
Trainer, TrainingArguments
from datasets import Dataset
import torch
from transformers import DataCollatorForLanguageModeling
from peft import get_peft_model, LoraConfig, TaskType
```

PART 04

Explore the Word Representations

Track 4.1 Visualise the word representations by t-SNE

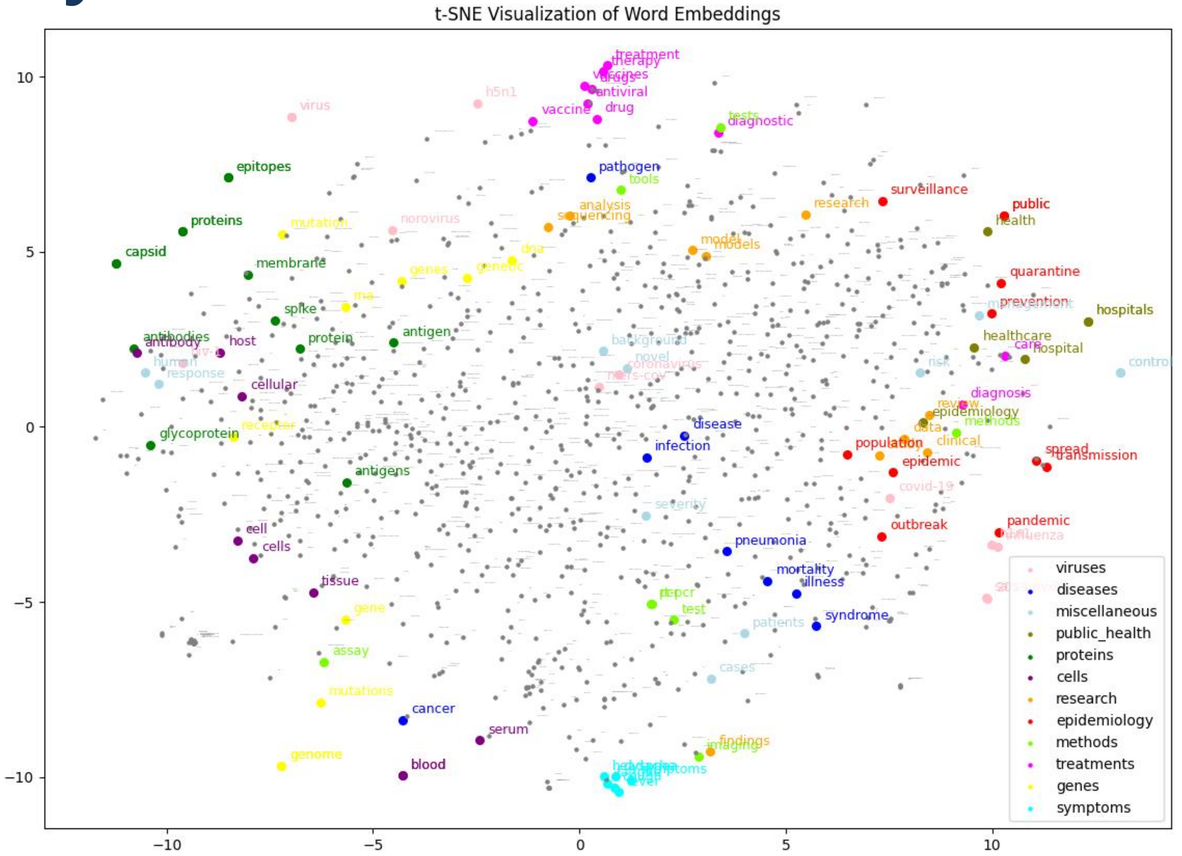
- **t-SNE** is a dimensionality reduction algorithm that visualizes high-dimensional data by mapping it into a lower-dimensional space (typically 2D or 3D) while preserving local structure and patterns.



Imperial DS&AI Winter School

Track 4.2 Visualise Biomedical Entities by t-SNE

- **Biomedical Entities** words are colored by category and highlighted in this section. In this graph, words marked with the same color tend to cluster together, while words marked with the different color tend to separate from each other.



Track 4.3 Co-occurrence

- The words are selected in order from highest to lowest according to the **statistical frequency** of co-occurrence with the target word in the corpus.

- **Cross-Domain Relevance**
- **Contextual Associations**
- **Broad Impact of COVID-19**

Co-occuring Word	Frequency
Covid	0.009722719
multiplicity	0.006220484
flaviviral	0.005976294
Lyme	0.005395208
productive	0.005248906
obstructive	0.005074524
mouth	0.004548549
persistent	0.004354533
Alzheimer	0.004330223
SFTSV	0.004047467

Target word: ‘coronavirus’

Table 4.4: 10 biomedical entities with the highest frequency of cooccurrence with coronavirus

Track 4.4 Semantic Similarity

- The words are selected in order from highest to lowest according to the **cosine similarity** of the word vector with the target words.

$$Similarity = \frac{\boldsymbol{v_1} \cdot \boldsymbol{v_2}}{\|\boldsymbol{v_1}\| \cdot \|\boldsymbol{v_2}\|}$$

- **Model Strengths**
- **Error Handling**
- **Temporal and Specific References**

Semantic Similar Word	Similarity
novel	0.632027924
2019-novel	0.618497908
coronovirus	0.610034168
abstract	0.596071839
provisionally	0.592935622
2019-ncov	0.58953917
2019	0.587587357
ncov-2019	0.576685071
cov	0.576638222
2019-novel	0.618497908

Target word: ‘coronavirus’

Table 4.3: 10 biomedical entities with the highest semantic similarity with coronavirus

The end

Thank you for your attention!

Reference：

[1] S. Bird, “Nltk: The natural language toolkit,” in Annual Meeting of the Association for Computational
Linguistics, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:1438450
[2] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, “Class-based n-gram models
of natural language,” Comput. Linguist., vol. 18, no. 4, p. 467–479, Dec. 1992.
[3] A. Fonarev, O. Hrinchuk, G. Gusev, P. Serdyukov, and I. Oseledets, “Riemannian optimization for
skip-gram negative sampling,” ArXiv, vol. abs/1704.08059, 2017.
[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep
bidirectional trans_x0002_formers for language understanding,” in North American
Chapter of the Association for Computa_x0002_tional Linguistics, 2019.
[5] J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, “Lora: Low-rank adaptation
of large language models,” ArXiv, vol. abs/2106.09685, 2021.