Tecnología al servicio de la salud:
Optimizando el tratamiento de pacientes con inteligencia artificial

Dr. Yamil Vindas-Yassine
Instituto Nacional de Investigación en Informática y Automática (INRIA), Lyon, Francia

Estructura

I. **Contexto**
   a) **Prevención de accidentes cerebrovasculares**
   b) **Otras aplicaciones de control médico**
   c) **Desafíos existentes**

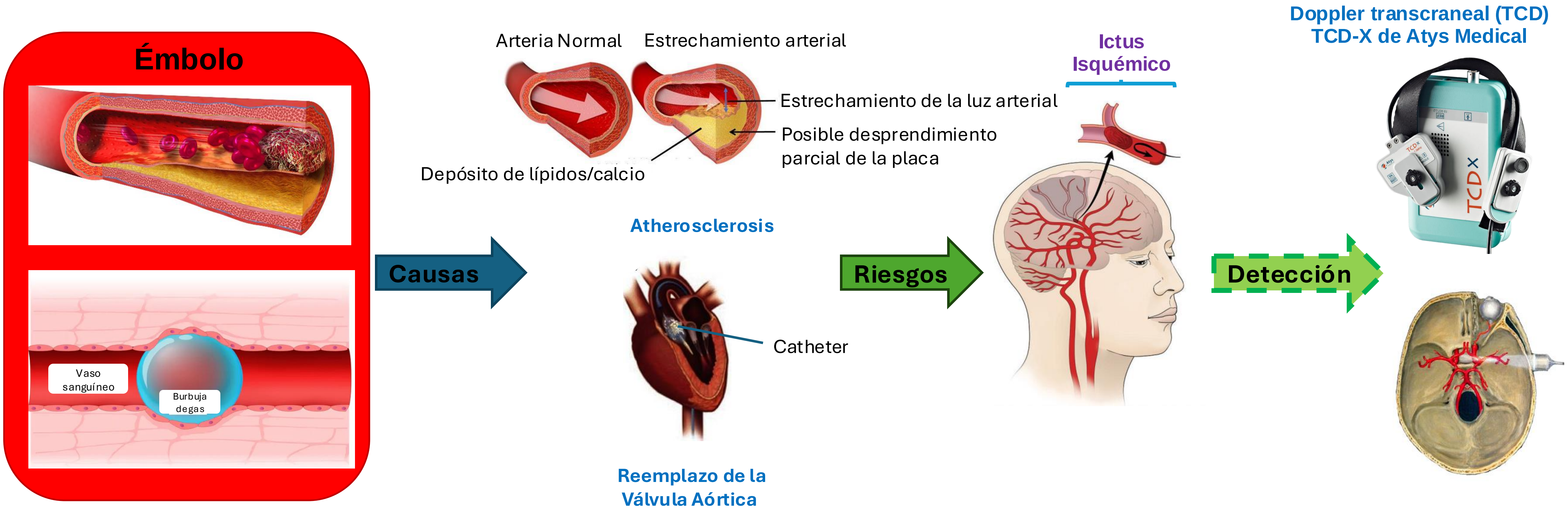
II. **Introducción al aprendizaje automático**
   a) **Tipos de aprendizaje**
   b) **Principio de entrenamiento**

III. **Inteligencia artificial para la medicina**
   a) **Anotación semiautomática de datos**
   b) **Modelos multi-representación**
   c) **Compresión de modelos**

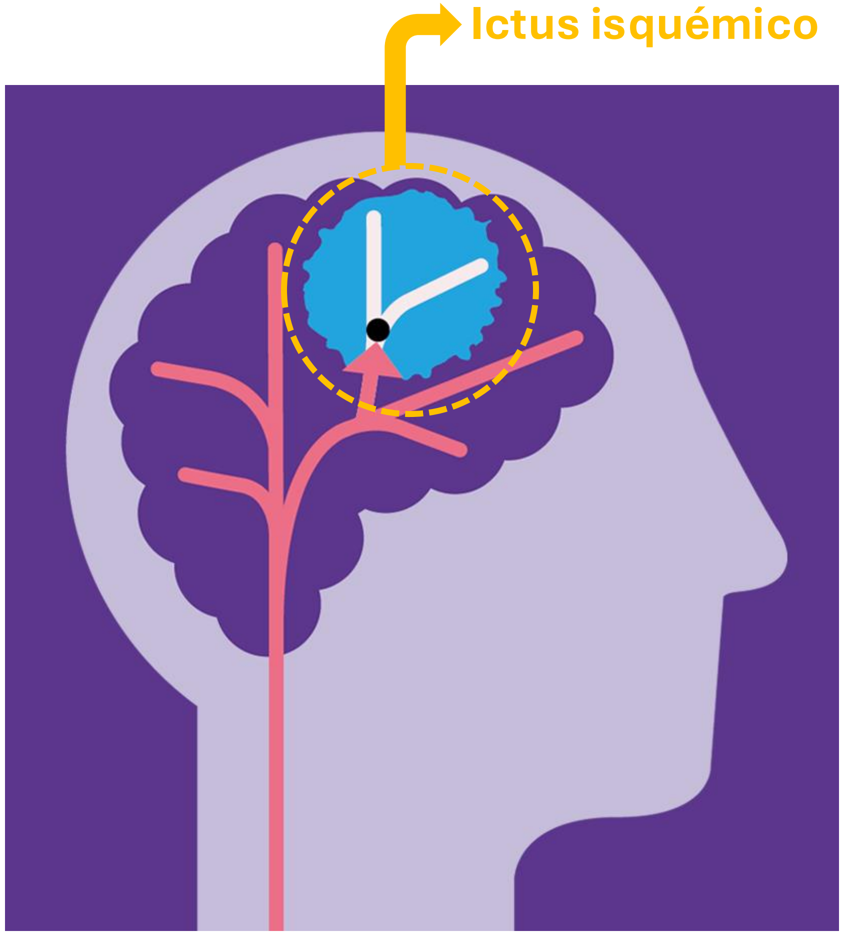
IV. **Conclusiones y perspectivas**

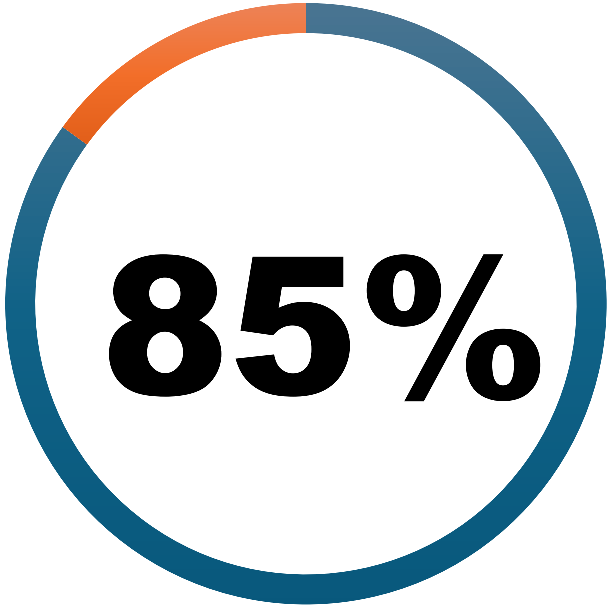
Estructura

Émbolos y accidentes cerebrovasculares (ACV)



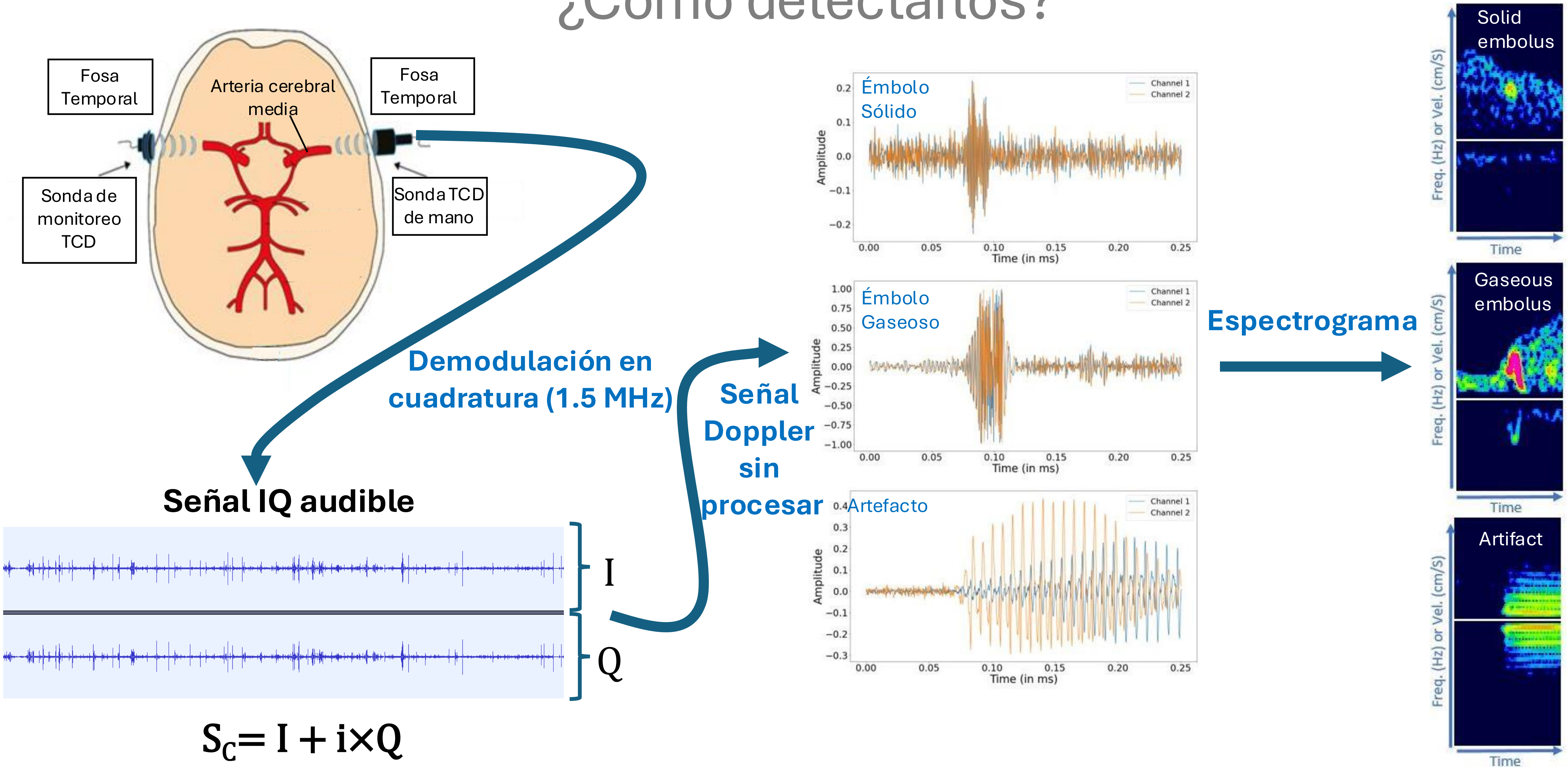
¿Por qué?





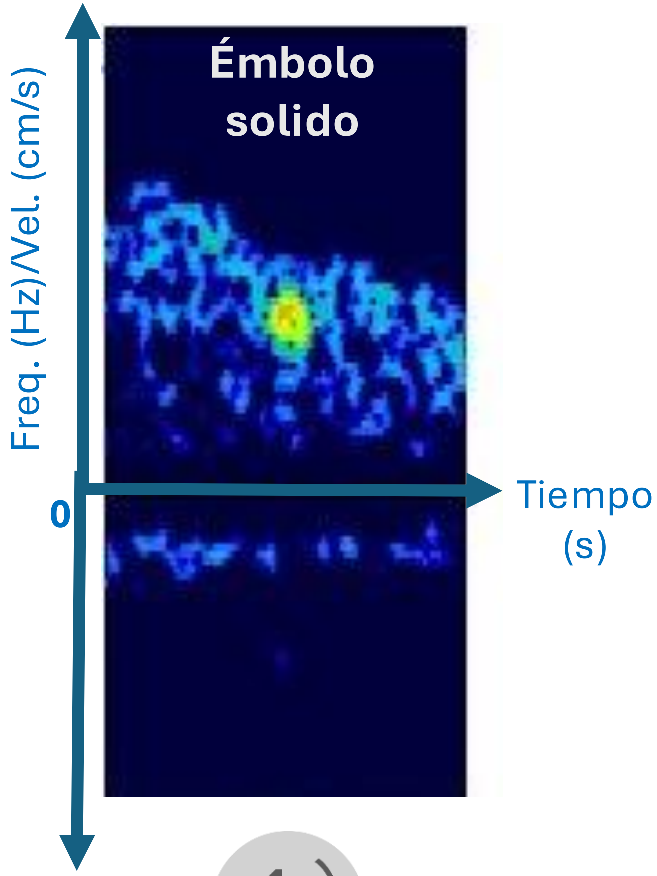
DE LOS ICTUS SE DEBEN A LA OBSTRUCCIÓN DE UNA ARTERIA CEREBRAL

¿Cómo detectarlos?



Criterios de detección

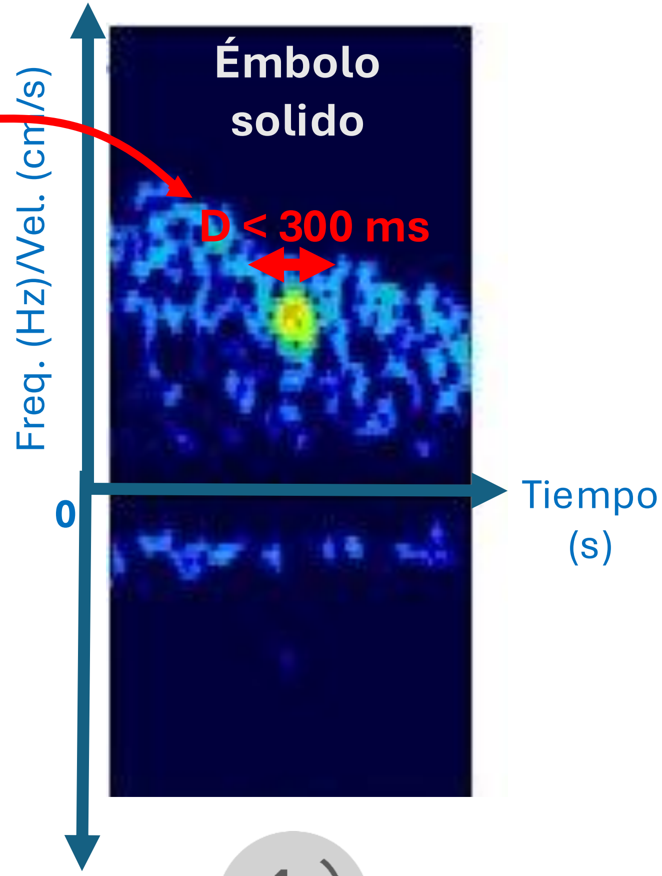
- **Criterios básicos de identificación de las señales microembólicas Doppler***
  - **Duración < 300 ms** ⟹ Señales transitorias de alta intensidad (**HITS**)
  - **Unidireccional** en el dominio tiempo-frecuencia.
    - ⟹ **No hay simetría** con respecto a la línea de base de frecuencia cero.
  - **Sonido musical** «chirrido» o «chasquido».
  - **Aumento de la intensidad** de al menos 3 dB con respecto a la señal de flujo sanguíneo
    - ⟹ Definido a través del “hits-to-blood ratio” (**HBR**)



*Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.

Criterios de detección

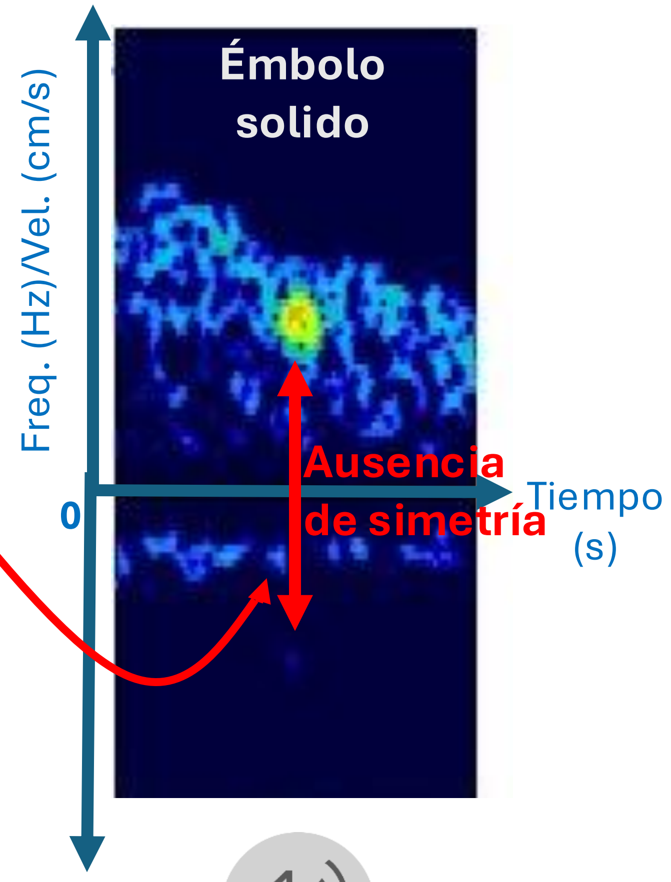
- **Criterios básicos de identificación de las señales microembólicas Doppler***
  - **Duración < 300 ms** ⟹ Señales transitorias de alta intensidad (**HITS**)
  - **Unidireccional** en el dominio tiempo-frecuencia.
    - ⟹ **No hay simetría** con respecto a la línea de base de frecuencia cero.
  - **Sonido musical** «chirrido» o «chasquido».
  - **Aumento de la intensidad** de al menos 3 dB con respecto a la señal de flujo sanguíneo
    - ⟹ Definido a través del “hits-to-blood ratio” (**HBR**)



*Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.

Criterios de detección

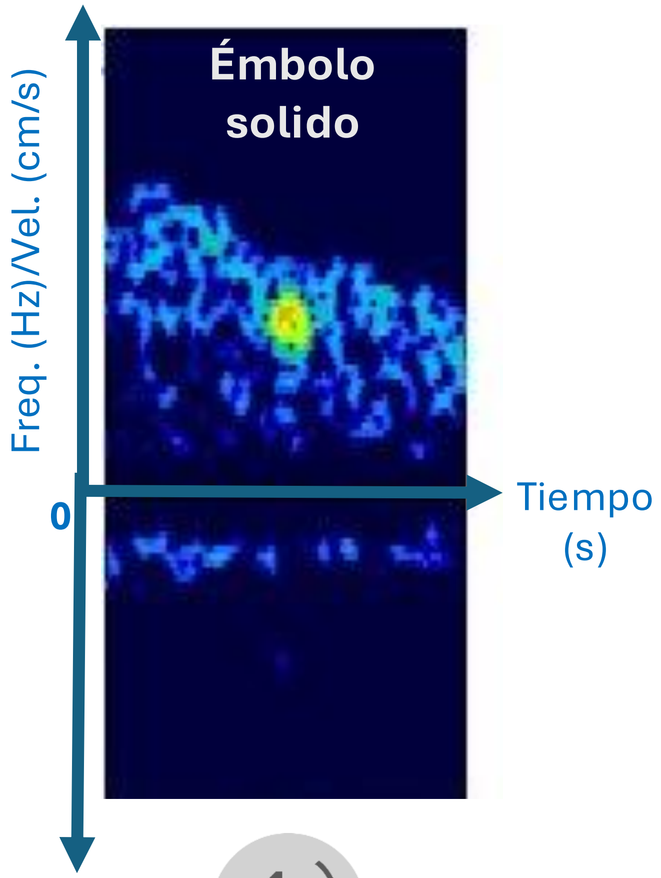
- **Criterios básicos de identificación de las señales microembólicas Doppler***
  - **Duración < 300 ms** ⟹ Señales transitorias de alta intensidad (**HITS**)
  - **Unidireccional** en el dominio tiempo-frecuencia.
    - ⟹ **No hay simetría** con respecto a la línea de base de frecuencia cero.
  - **Sonido musical** «chirrido» o «chasquido».
  - **Aumento de la intensidad** de al menos 3 dB con respecto a la señal de flujo sanguíneo
    - ⟹ Definido a través del “hits-to-blood ratio” (**HBR**)



*Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.

Criterios de detección

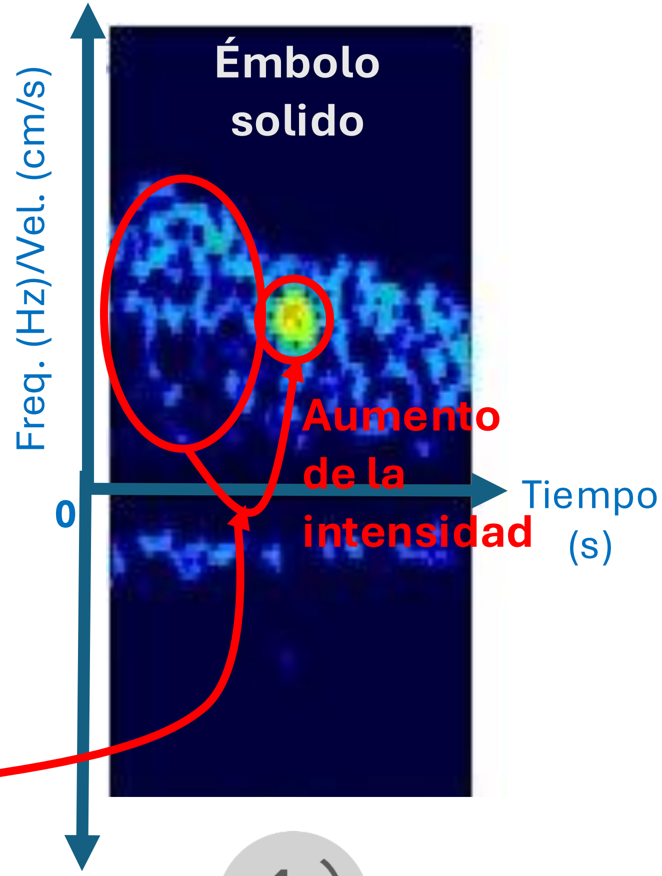
- Criterios básicos de identificación de las señales microembólicas Doppler*
  - **Duración < 300 ms** ⟹ Señales transitorias de alta intensidad (**HITS**)
  - **Unidireccional** en el dominio tiempo-frecuencia.
    - ⟹ **No hay simetría** con respecto a la línea de base de frecuencia cero.
  - **Sonido musical** «chirrido» o «chasquido».
  - **Aumento de la intensidad** de al menos 3 dB con respecto a la señal de flujo sanguíneo
    - ⟹ Definido a través del “hits-to-blood ratio” (**HBR**)



*Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.

Criterios de detección

- **Criterios básicos de identificación de las señales microembólicas Doppler***
  - **Duración < 300 ms** ⟹ Señales transitorias de alta intensidad (**HITS**)
  - **Unidireccional** en el dominio tiempo-frecuencia.
    - ⟹ **No hay simetría** con respecto a la línea de base de frecuencia cero.
  - **Sonido musical** «chirrido» o «chasquido».
  - **Aumento de la intensidad** de al menos 3 dB con respecto a la señal de flujo sanguíneo
    - ⟹ Definido a través del "hits-to-blood ratio" (**HBR**)



*Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.

Estructura

I. **Contexto**
   a) Prevención de accidentes cerebrovasculares
   b) **Otras aplicaciones de control médico**
   c) Desafíos existentes

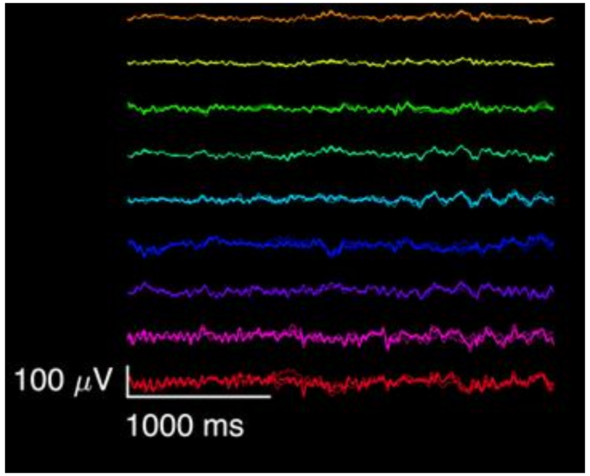
II. Introducción al aprendizaje automático
   a) Tipos de aprendizaje
   b) Principio de entrenamiento

III. Inteligencia artificial para la medicina
   a) Anotación semiautomática de datos
   b) Modelos multi-representación
   c) Compresión de modelos

IV. Conclusiones y perspectivas

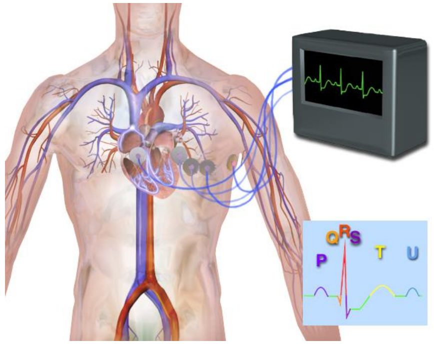
Otras aplicaciones de monitoreo

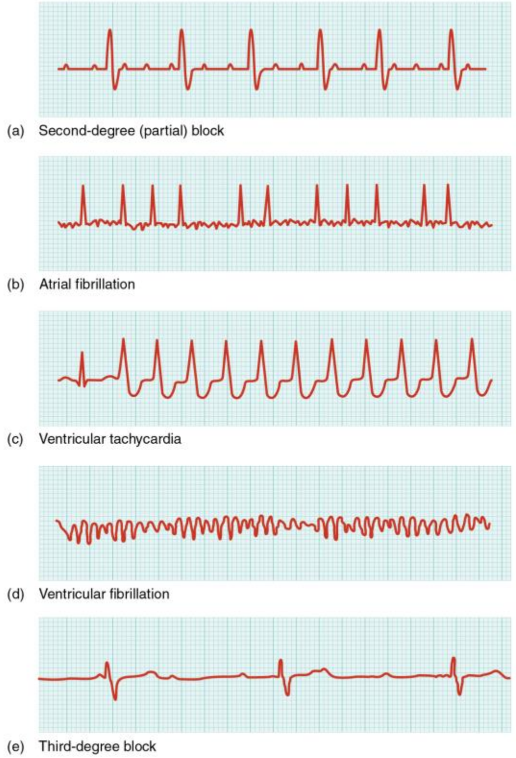
Reconocimiento de crisis epilépticas



Electroencefalograma

Detección y clasificación de arritmias





Electrocardiograma

Estructura

I. **Contexto**
   a) Prevención de accidentes cerebrovasculares
   b) Otras aplicaciones de control médico
   c) **Desafíos existentes**

II. Introducción al aprendizaje automático
   a) Tipos de aprendizaje
   b) Reducción de dimensión
   c) Principio de entrenamiento

III. Inteligencia artificial para la medicina
   a) Anotación semiautomática de datos
   b) Modelos multi-representación
   c) Compresión de modelos

IV. Conclusiones y perspectivas

Retos: anotación de datos

- Dificultad para encontrar conjuntos de datos públicos
- Dificultad de anotación→ Etiquetas con ruido
- Anotación costosa (8685/68491 muestras etiquetadas).
- Clases desequilibradas (émbolos sólidos < 10% HITS).

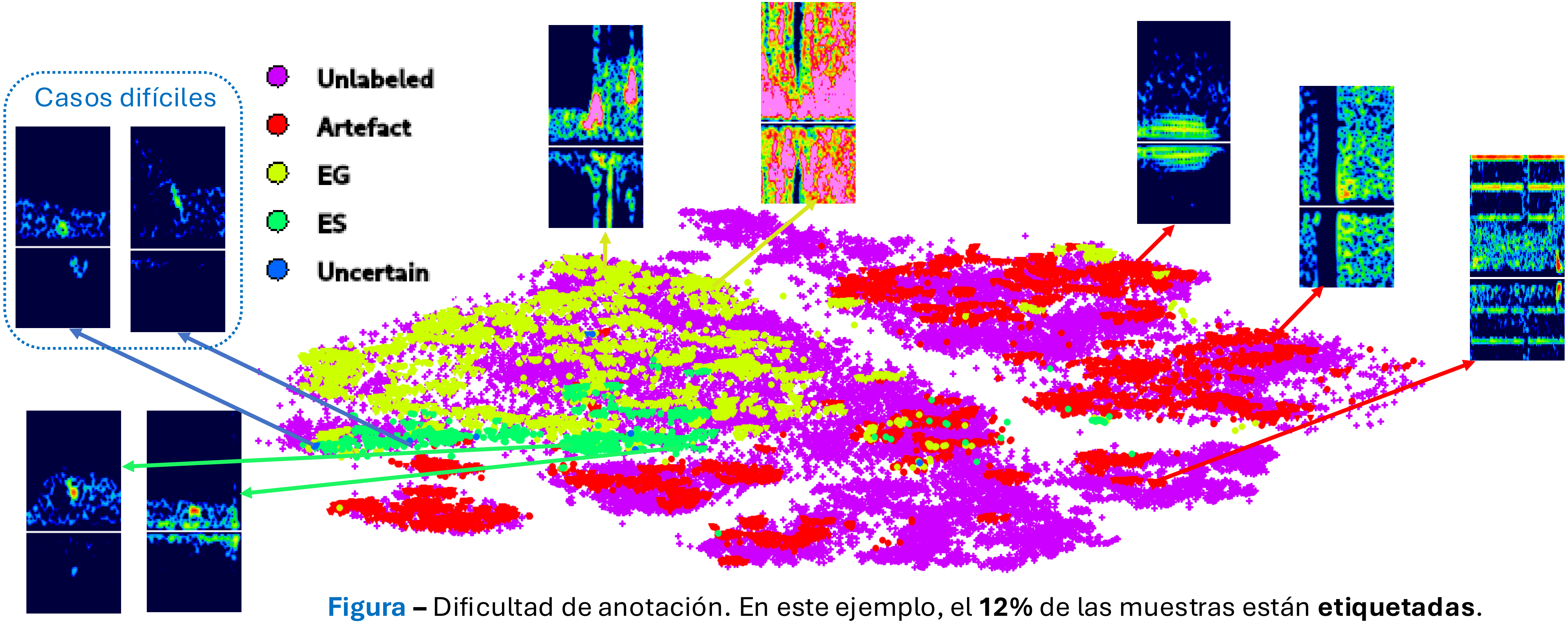
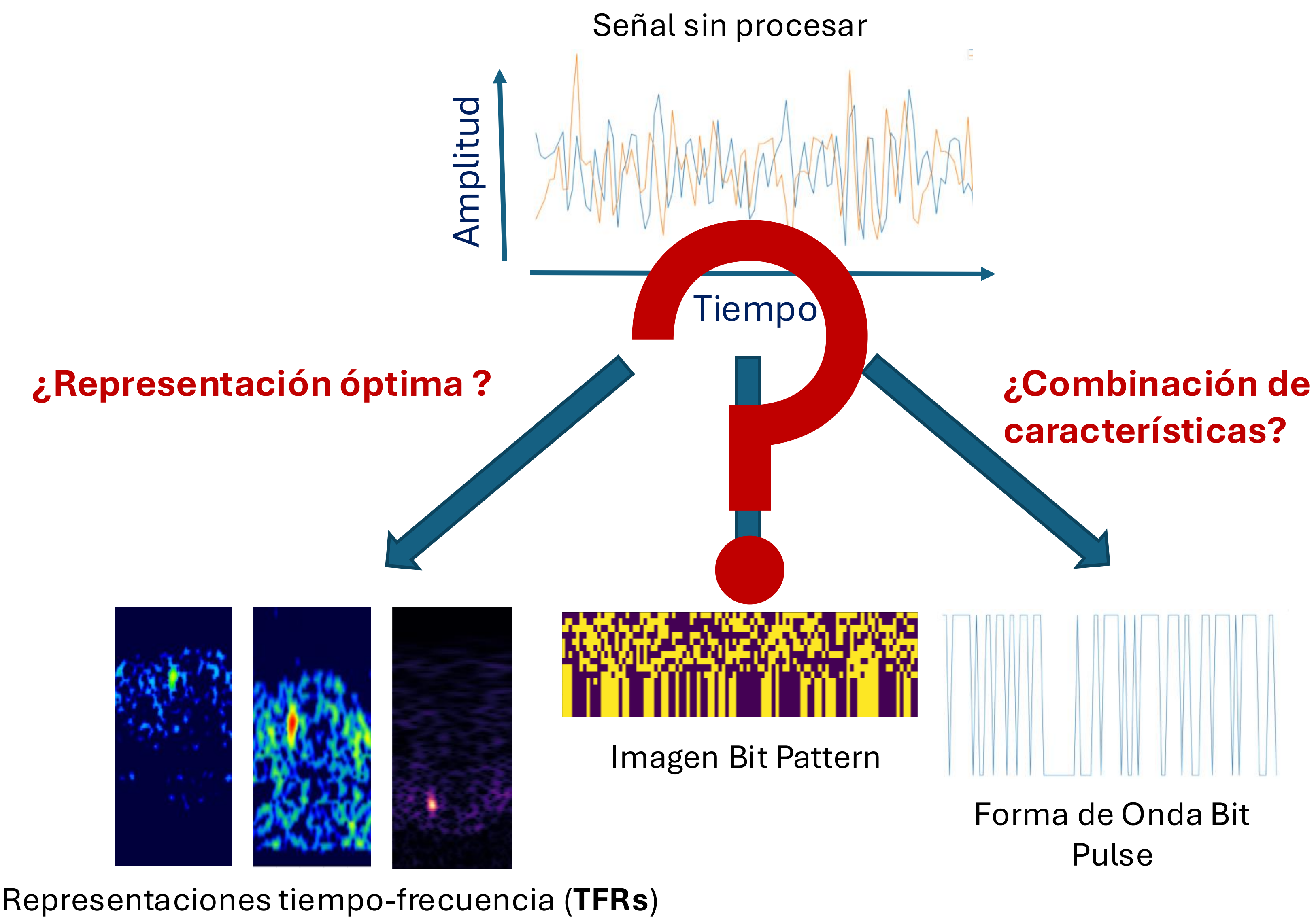


Figura – Dificultad de anotación. En este ejemplo, el **12%** de las muestras están **etiquetadas**.

Retos: representación óptima



Retos: compresión de modelos

Figura – Doppler transcraneal portátil (TCD) de Atys Medical.

- Recursos de memoria limitados.
- Recursos de cálculo limitados.
- Limitaciones energéticas.

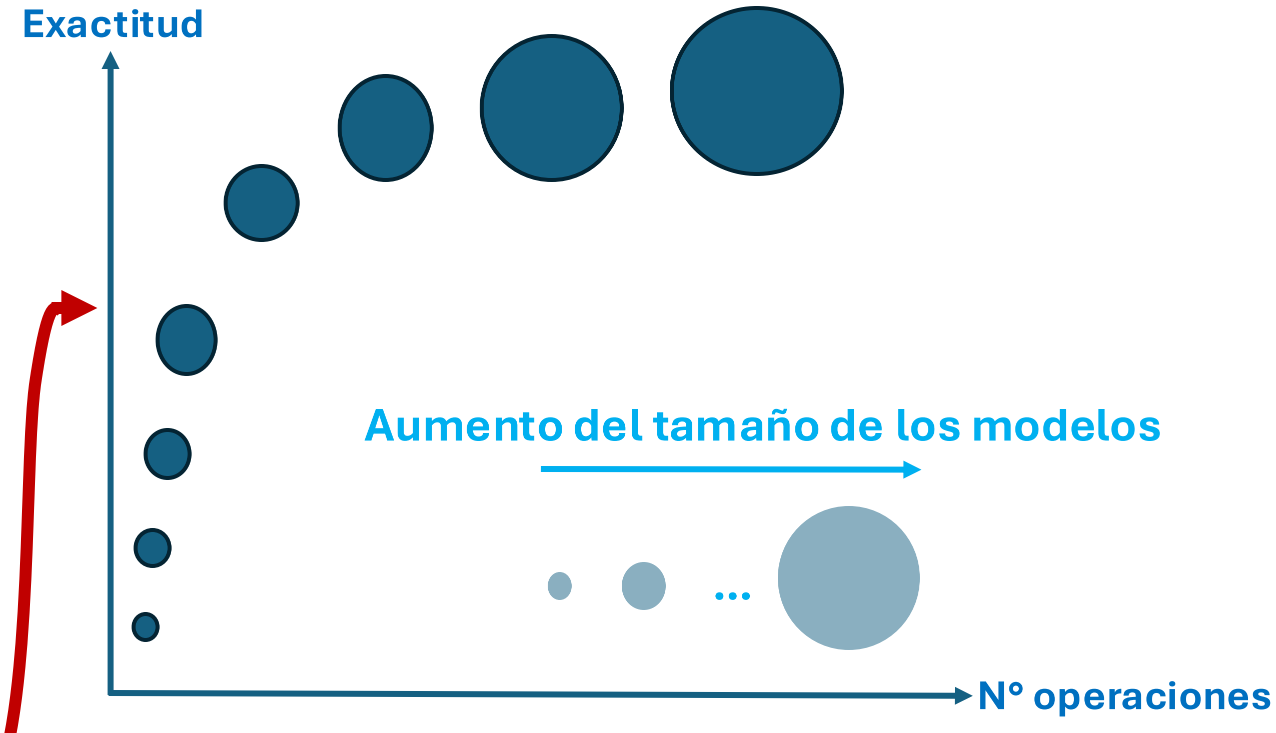
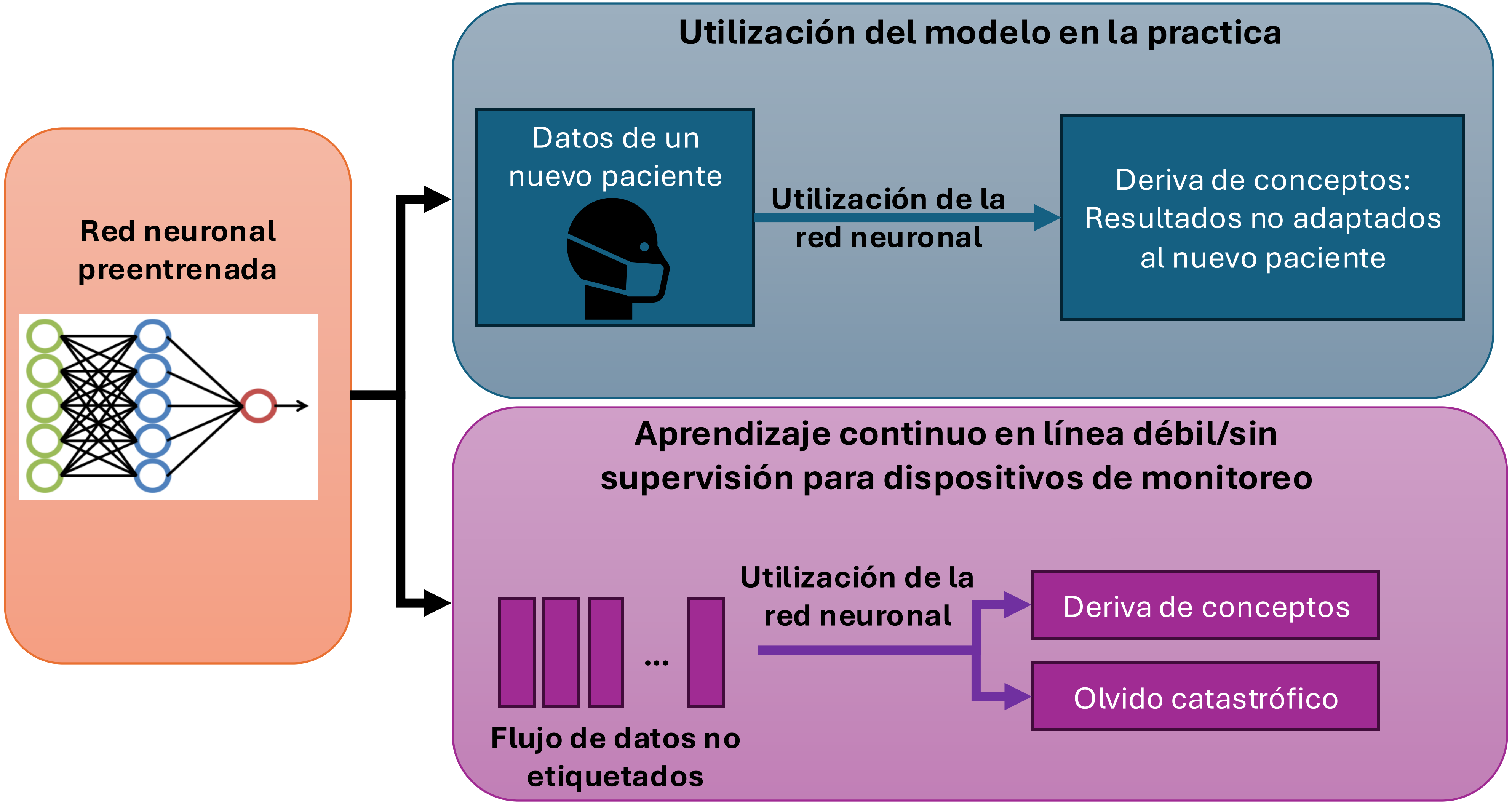


Figura – Exactitud de clasificación basada en el tamaño y el número de operaciones en coma flotante de diferentes modelos de aprendizaje profundo (inspirado en Abbas et al. 2021).

Retos: aprendizaje continuo



Posibles soluciones

Creación y anotación de conjuntos de datos

- Anotación de datos semi-supervisada*
- Etiquetado flexible (anotación)*

Múltiples representaciones

- Diferentes modelos con diferentes entradas**
- Modelo multi-representación

Modelos con gran demanda de recursos

- Modelos ligeros
- Compresión de modelos***
- (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

Estructura

I. Contexto
   a) Prevención de accidentes cerebrovasculares
   b) Otras aplicaciones de control médico
   c) Desafíos existentes

II. **Introducción al aprendizaje automático**
   a) **Tipos de aprendizaje**
   b) Reducción de dimensión
   c) Principio de entrenamiento

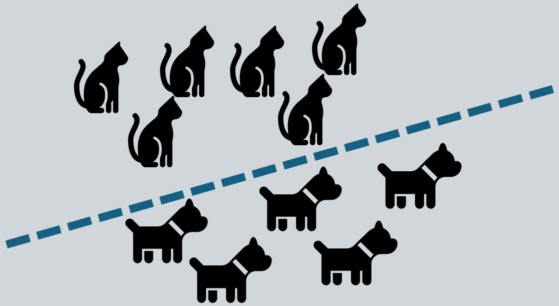
III. Inteligencia artificial para la medicina
   a) Anotación semiautomática de datos
   b) Modelos multi-representación
   c) Compresión de modelos

IV. Conclusiones y perspectivas

Principales tipos de aprendizaje

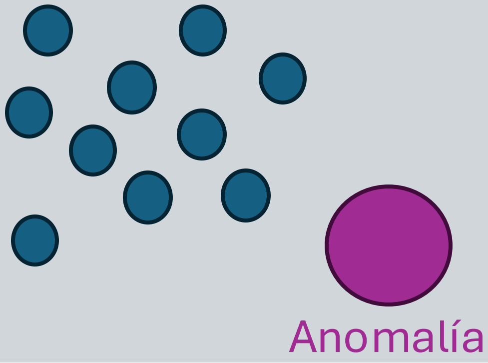
Supervisado

- **Descripción**: aprende viendo ejemplos con etiquetas
- **Ejemplo**: clasificación de arritmias.



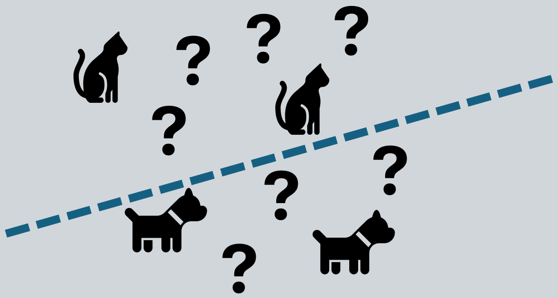
No supervisado

- **Descripción**: aprende la estructura de los datos sin etiquetas.
- **Ejemplo**: detección de anomalías cerebrales.



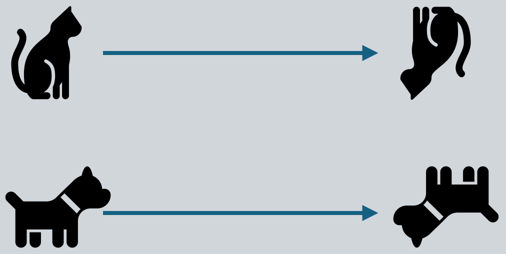
Semi-supervisado

- **Descripción**: aprende viendo pocos ejemplos sin etiquetas y muchos con.
- **Ejemplo**: clasificación de émbolos cerebrales.



Auto-supervisado

- **Descripción**: aprende con etiquetas generadas automáticamente.
- **Ejemplo**: análisis de registros médicos electrónicos.



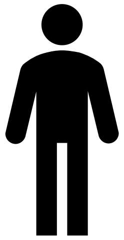
Estructura

Reducción de dimensión

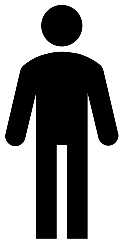
Principio: **escoger** las **características más importantes** para describir las entidades que queremos.

Nombre: Dani
Edad: 3 meses
Altura: 58 cm

Nombre: Val
Edad: 7 años
Altura: 110 cm



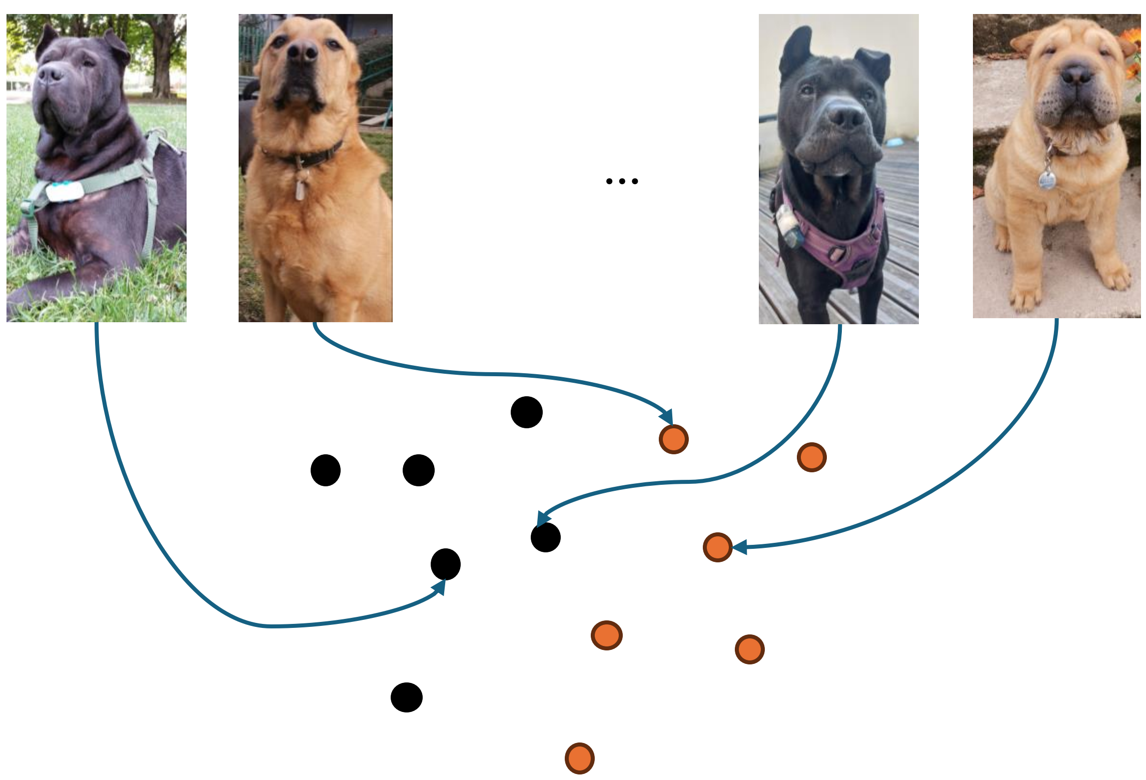
Nombre: Luci
Edad: 29 años
Altura: 170 cm



Nombre: Aquil
Edad: 58 años
Altura: 178 cm

Nombre: Reni
Edad: 5 años
Altura: 107 cm

Imágenes de perros



Puntos representando diferentes perros

Estructura

I. Contexto
   a) Prevención de accidentes cerebrovasculares
   b) Otras aplicaciones de control médico
   c) Desafíos existentes

II. **Introducción al aprendizaje automático**
   a) Tipos de aprendizaje
   b) Reducción de dimensión
   c) **Principio de entrenamiento**

III. Inteligencia artificial para la medicina
   a) Anotación semiautomática de datos
   b) Modelos multi-representación
   c) Compresión de modelos

IV. Conclusiones y perspectivas

Principio de entrenamiento

Objetivo: Enseñarle a un perro a reconocer un hueso de un balón de futbol. Tiene que hacer la menor cantidad de errores posibles.

Entrenamiento: Mostrarle varios balones y huesos de diferentes tamaños, formas, colores, etc., diciéndole cada vez a que corresponde cada objeto.

Prueba: Una vez que él perro vio cierta cantidad de objetos, mostrarle un objeto sin decirle que es y el perro tiene que apretar un botón si es un hueso y otro si es un balón. Cada vez que se equivoca, se corrige al perro.

Principio de entrenamiento

- Perro → **Modelo**
- Hueso y balón → **Datos** (imágenes, señales, texto, ...)
- Indicación sobre el tipo del objeto → **Etiqueta**
- Objetivo y corrección de las respuestas incorrectas → **Función pérdida**

Posibles soluciones

Creación y anotación de conjuntos de datos

- Anotación de datos semi-supervisada*
- Etiquetado flexible (anotación)*

Múltiples representaciones

- Diferentes modelos con diferentes entradas**
- Modelo multi-representación

Modelos con gran demanda de recursos

- Modelos ligeros
- Compresión de modelos***
- (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

Estructura

Posibles soluciones

Creación y anotación de conjuntos de datos →

- Anotación de datos semi-supervisada*
- Etiquetado flexible (anotación)*

Múltiples representaciones →

- Diferentes modelos con diferentes entradas**
- Modelo multi-representación

Modelos con gran demanda de recursos →

- Modelos ligeros
- Compresión de modelos***
- (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

Diferentes etapas

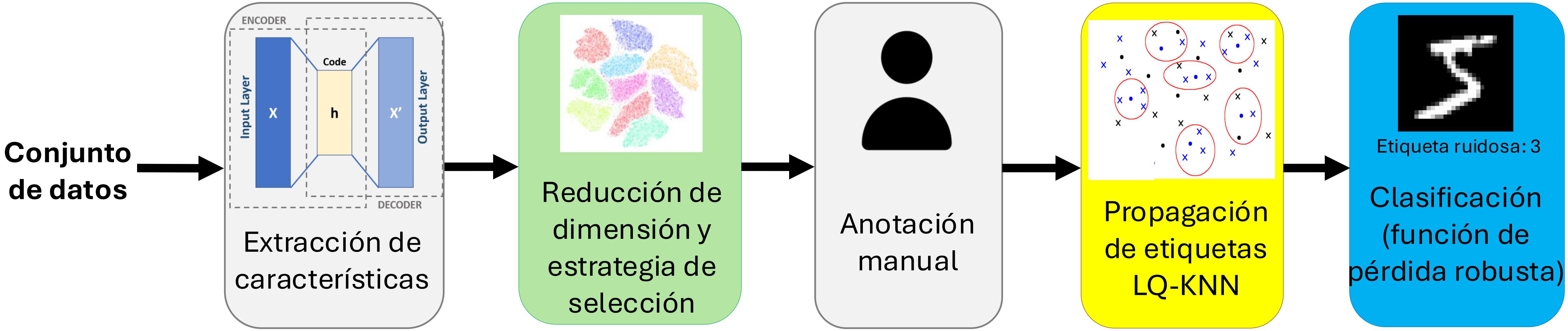


Figura - Método semiautomático de anotación de datos

Diferentes etapas

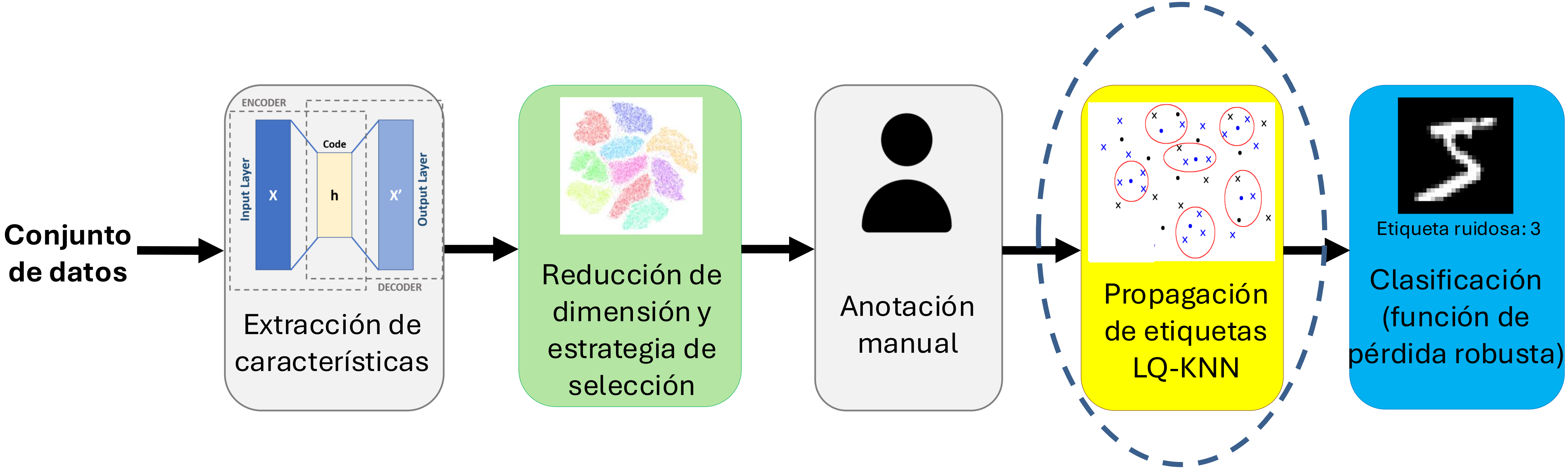


Figura - Método semiautomático de anotación de datos

Propagación semiautomática de etiquetas LQ-KNN

Conjunto de datos inicial

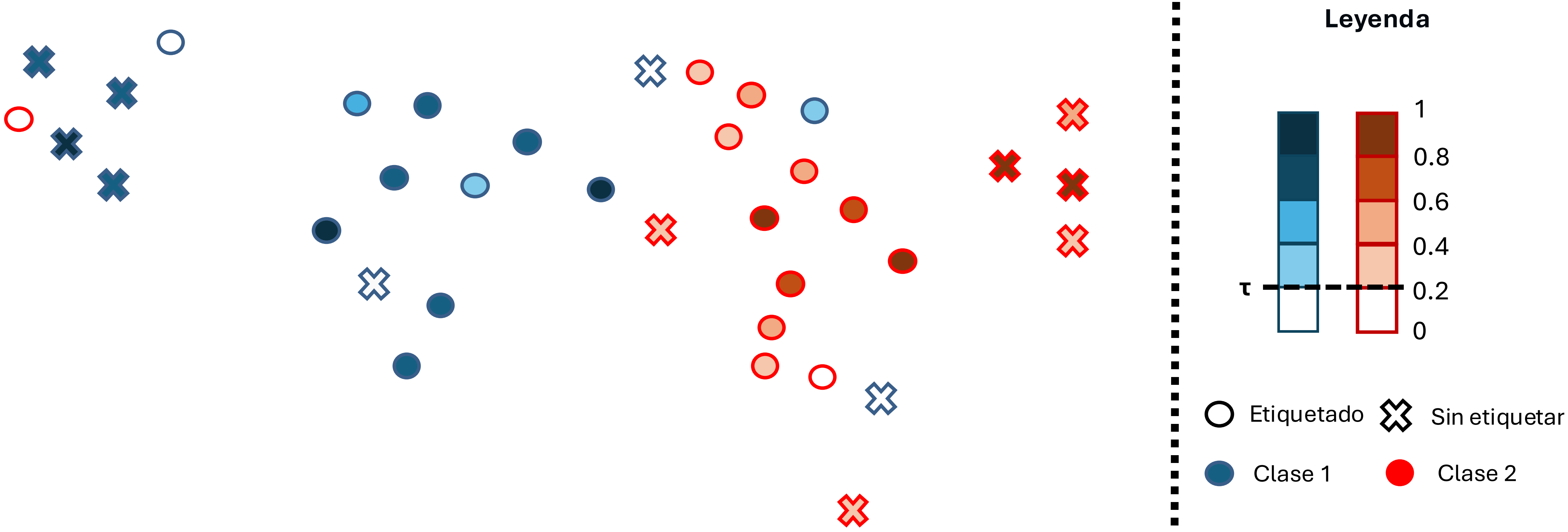


Figure – Ejemplo con dos vecinos (es decir, **K = 2** y **τ = 0,2**).

Propagación semiautomática de etiquetas LQ-KNN

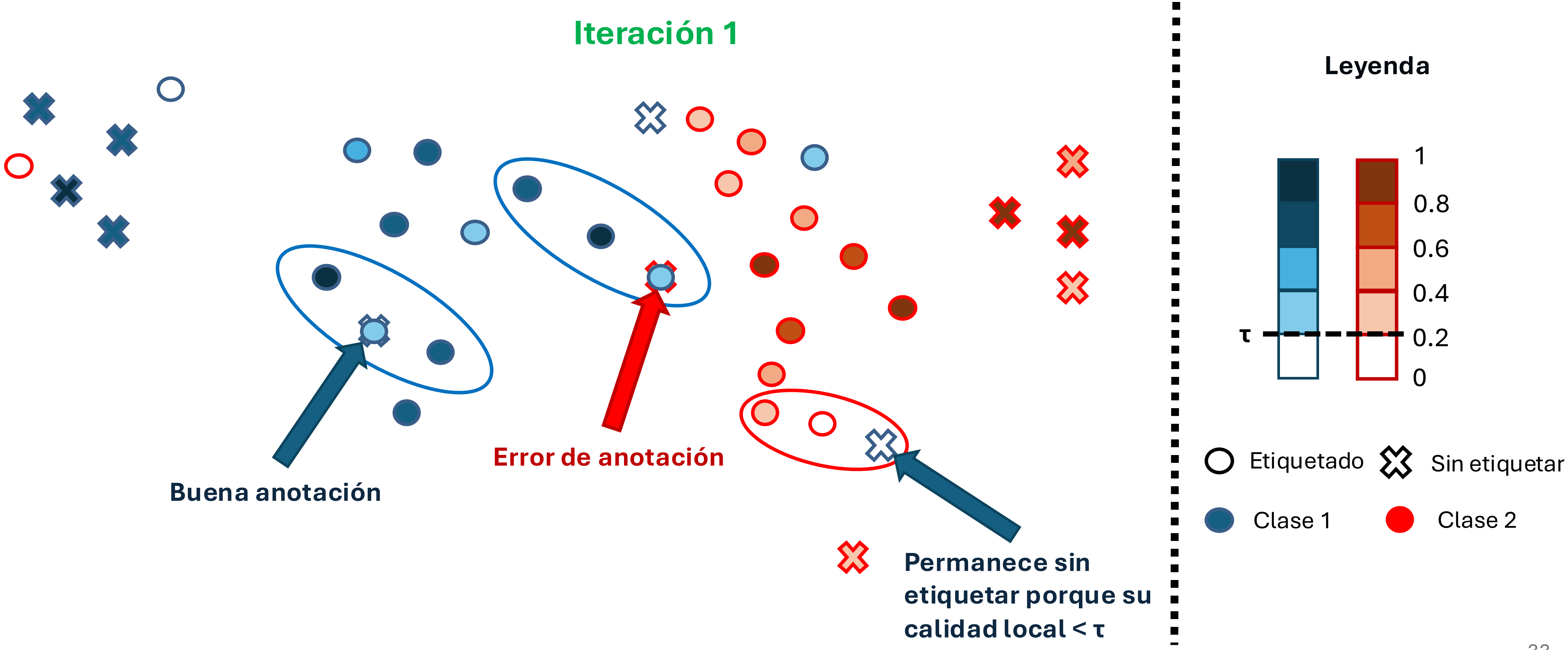


Figure – Ejemplo con dos vecinos (es decir, **K = 2** y **τ = 0,2**).

Propagación semiautomática de etiquetas LQ-KNN

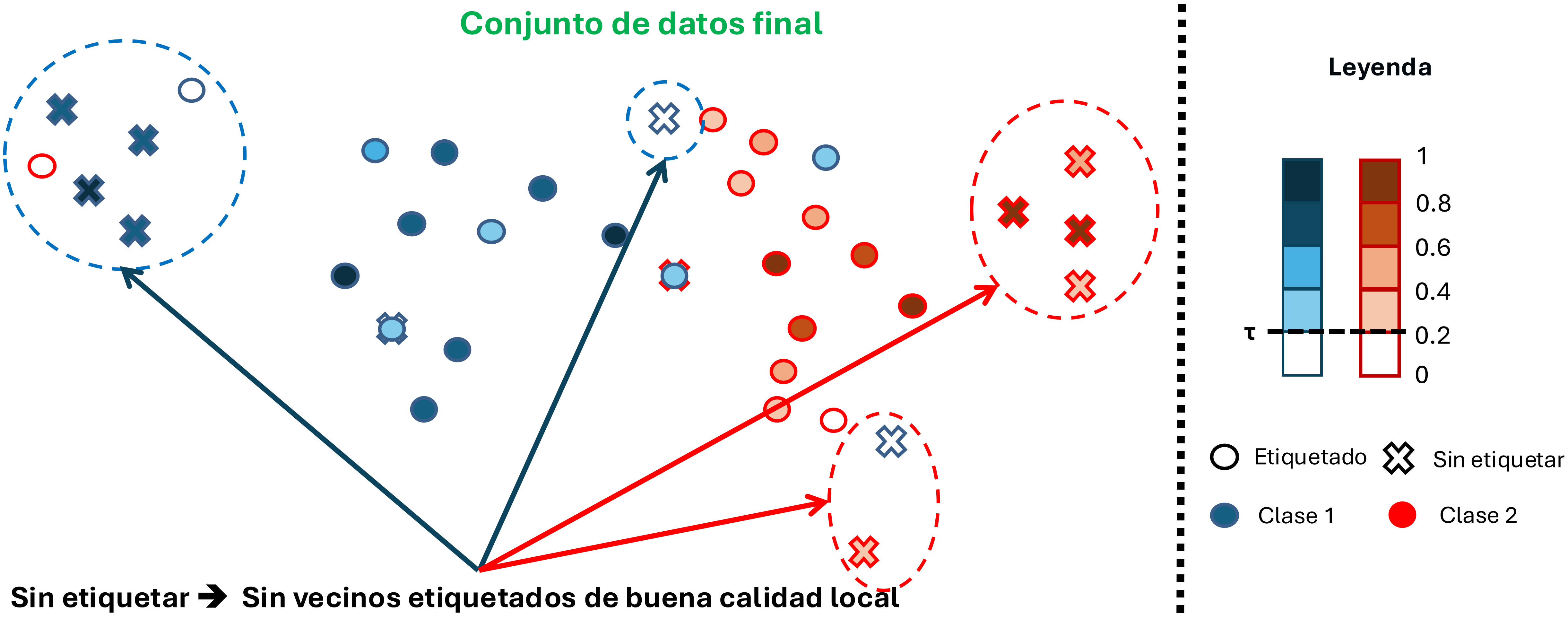


Figure – Ejemplo con dos vecinos (es decir, **K = 2** y **τ = 0,2**).

Experimento

Objetivo:

- Destacar la ventaja de la anotación semiautomática.
- Mostrar el impacto en la mejora del desempeño de los modelos.

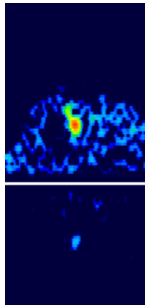
Medidas de desempeño:

- Tasa de error de anotación.
- Coeficiente de correlación de Mathews (MCC).
  - ➔ Permite de medir el desempeño del modelo, similar a la precisión o exactitud de clasificación.

Base de datos:

HITS:

- Datos TCD.
- 6 8491 imágenes.
- 1 545 manual-mente anotadas
- Tres clases.
- Frecuencia de muestreo: 4385 Hz.



Clase	Cantidad de muestras
Artefactos	403
Émbolos gaseosos	569
Émbolos solidos	569
Desconocido	4

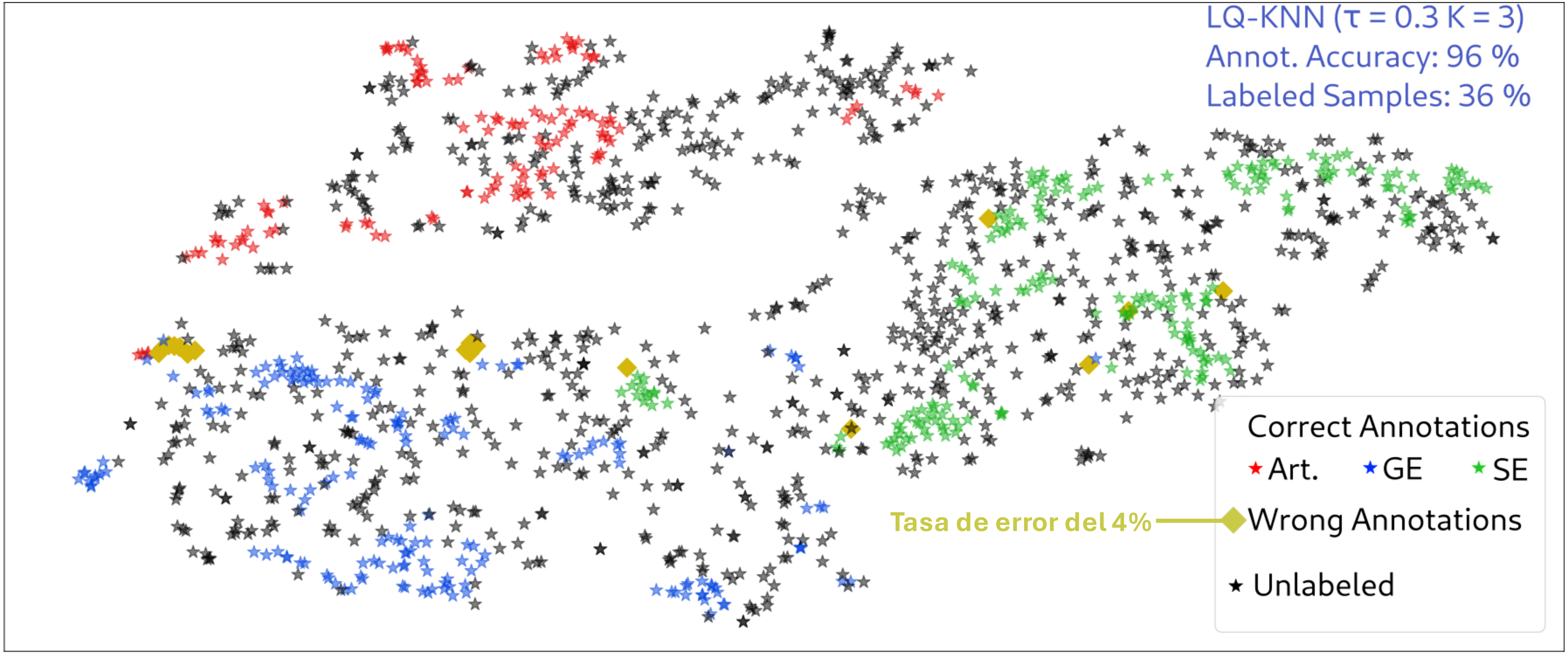


Figura - Propagación de etiquetas *LQ-KNN* con **K = 3** y **τ = 0.3**

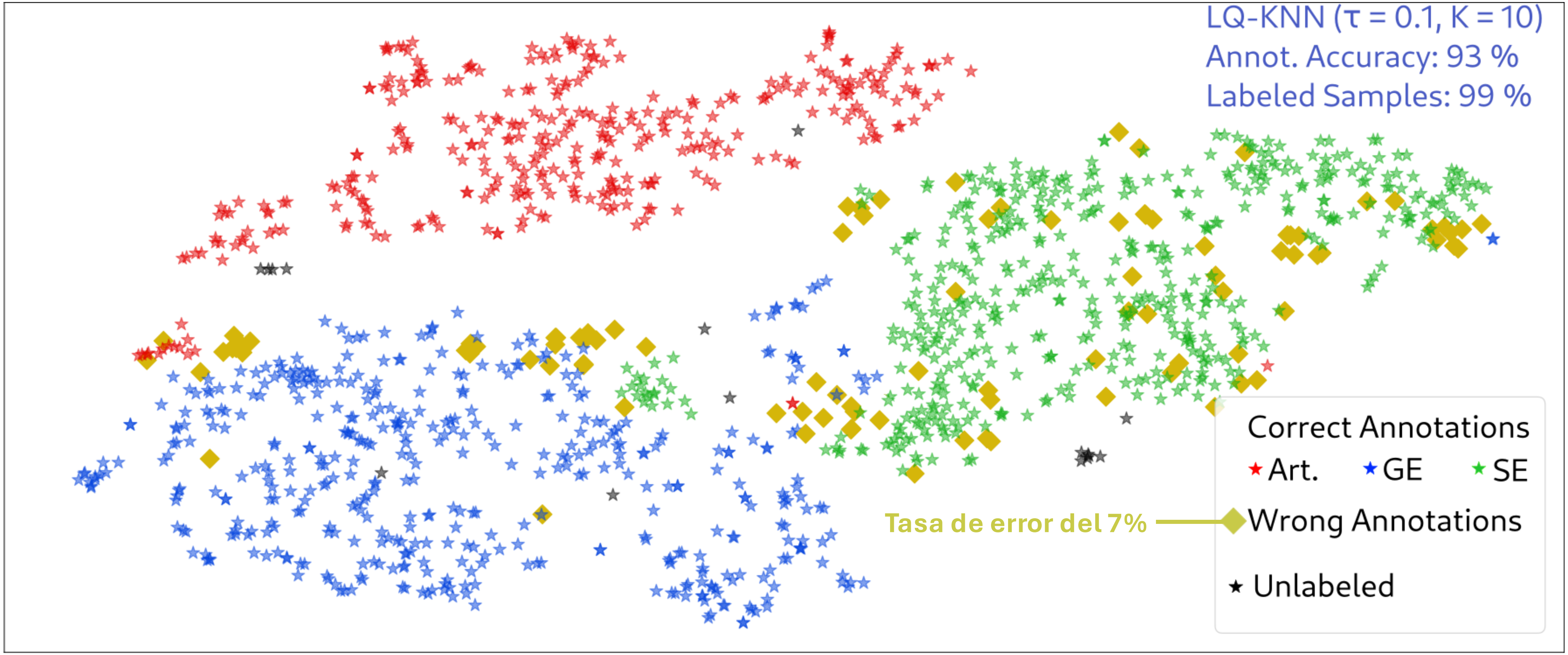


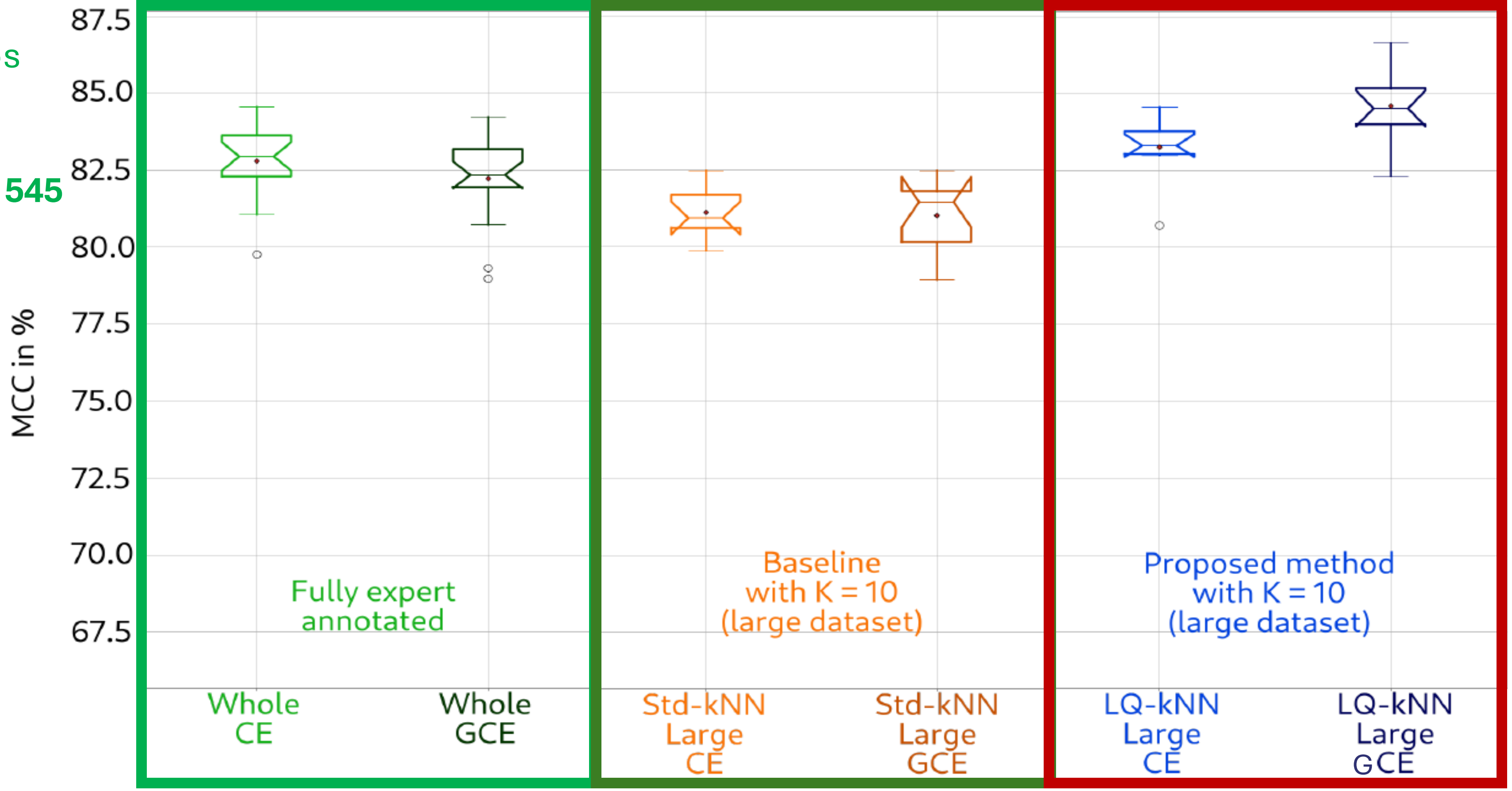
Figura - Propagación de etiquetas *LQ-KNN* con **K = 10** y **τ = 0.1**

K y **τ** controlan el **equilibrio** entre los **errores de anotación** y la **cantidad de datos etiquetados**.

Tabla - Resultados de clasificación de émbolos cerebrales

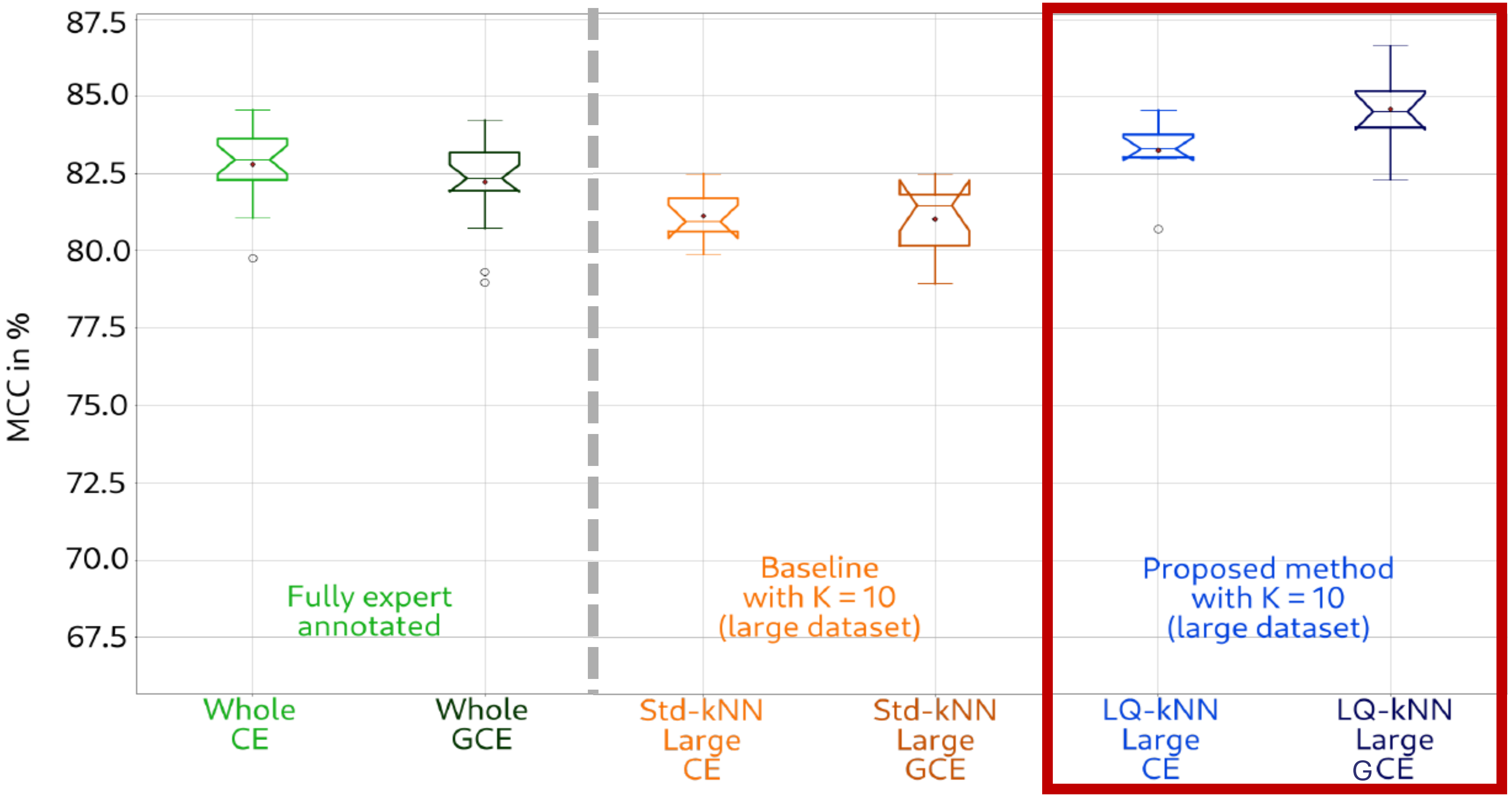
Conjunto de datos compuesto por **13 653 datos etiquetados: 1 545 etiquetados manualmente** y **12 108 etiquetados** con **Std-KNN**

Conjunto de datos HITS **etiquetado manualmente** compuesto por **1 545 datos**



Conjunto de datos compuesto por **14 970 datos etiquetados: 1 545** etiquetados **manualmente** y **13 425 etiquetados con** nuestro **LQ-KNN**

Tabla - Resultados de clasificación de émbolos cerebrales



Mejora del 1,78 % con respecto al conjunto de datos original

Estructura

I. Contexto
   a) Prevención de accidentes cerebrovasculares
   b) Otras aplicaciones de control médico
   c) Desafíos existentes

II. Introducción al aprendizaje automático
   a) Tipos de aprendizaje
   b) Reducción de dimensión
   c) Principio de entrenamiento

III. **Inteligencia artificial para la medicina**
   a) Anotación semiautomática de datos
   b) **Modelos multi-representación**
   c) Compresión de modelos

IV. Conclusiones y perspectivas

Posibles soluciones

Creación y anotación de conjuntos de datos

- Anotación de datos semi-supervisada*
- Etiquetado flexible (anotación)*

Múltiples representaciones

- Diferentes modelos con diferentes entradas**
- Modelo multi-representación

Modelos con gran demanda de recursos

- Modelos ligeros
- Compresión de modelos***
- (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

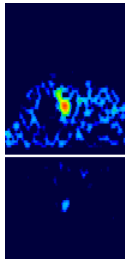
Método

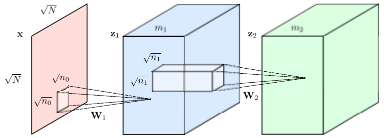
Objetivos:

- **Mejorar** la **clasificación** de las señales **TCD** para la **prevención** de **ACV**.
- **Aprovechar** la **complementariedad** de las distintas **representaciones**.

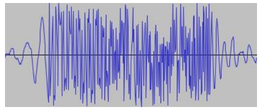
Modelos:

- Red Neuronal (CNN) 2D para las TFRs.

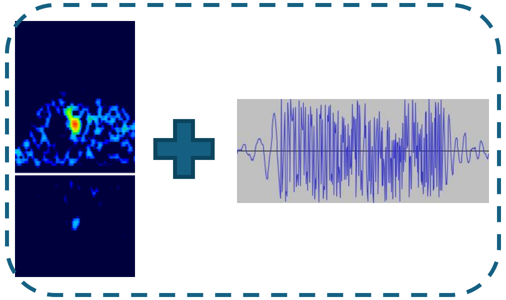


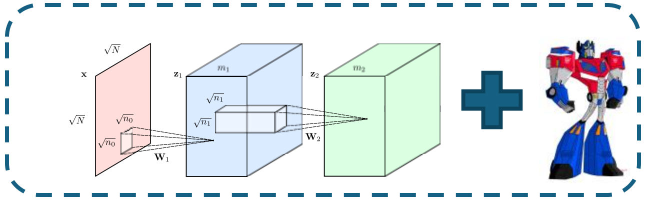


- Red Neuronal 1DD CNN-Transformer para las señales sin procesar.



- **Modelos híbridos para ambas representaciones.**





Experimento

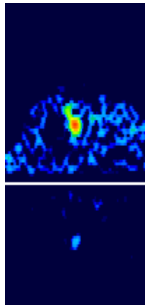
Objetivos:

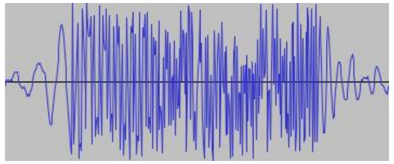
- Destacar la ventaja de usa múltiples representaciones.

Base de datos:

HITS:

- Datos TCD.
- 1 545 **imágenes** y **señales** sin procesar manualmente anotados
- Tres clases.
- Frecuencia de muestreo: 4385 Hz.



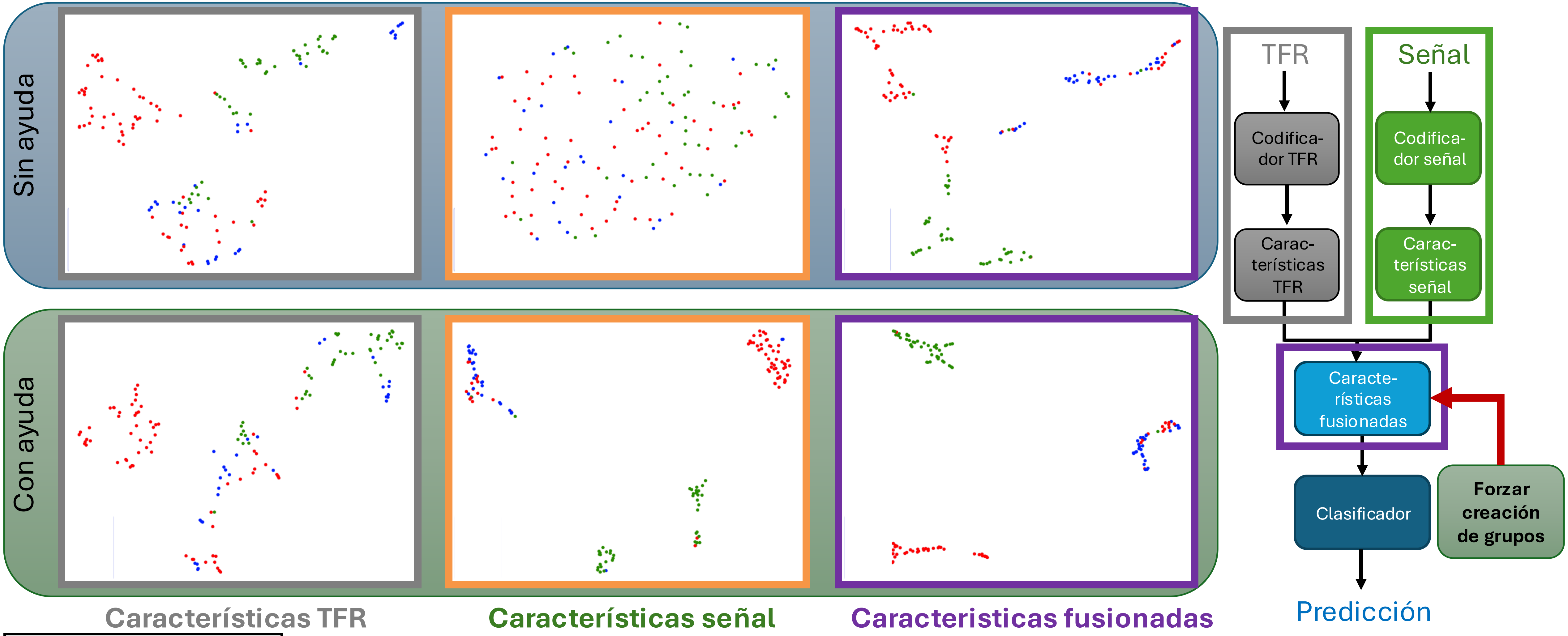


Medidas de desempeño:

- Coeficiente de correlación de Mathews (MCC).
  - ➔ Permite de medir el desempeño del modelo, similar a la precisión o exactitud de clasificación.

Clase	Cantidad de muestras
Artefactos	403
Émbolos gaseosos	569
Émbolos solidos	569

Resultados



Artefacto
Émbolo gaseoso
Émbolo solido

Figura – Representación 2D de las características que aprendió el modelo por cada entrada (TFR, señal y fusión)

Estructura

Posibles soluciones

Creación y anotación de conjuntos de datos

- Anotación de datos semi-supervisada*
- Etiquetado flexible (anotación)*

Múltiples representaciones

- Diferentes modelos con diferentes entradas**
- Modelo multi-representación

Modelos con gran demanda de recursos

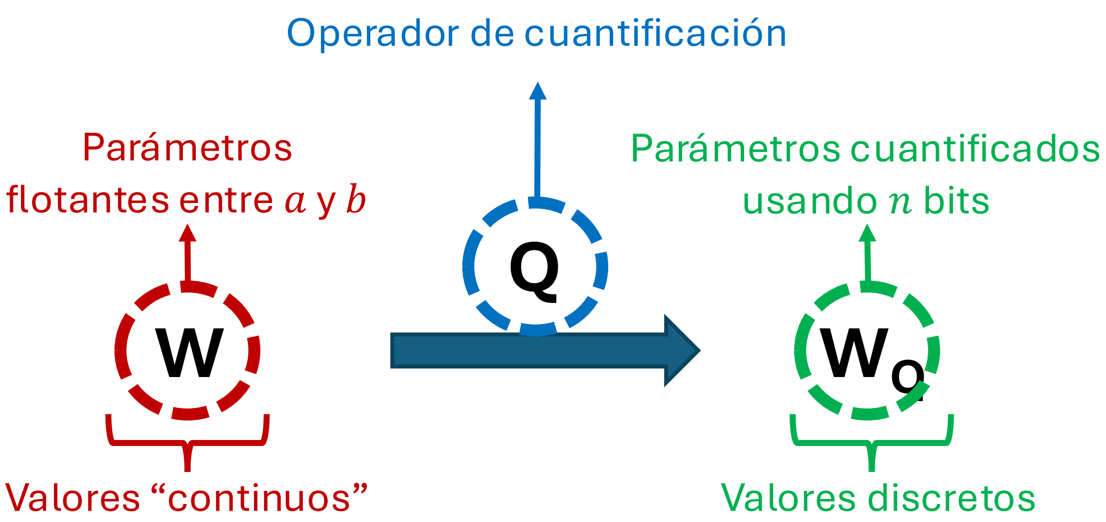
- Modelos ligeros
- Compresión de modelos***
- (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

Cuantificación y poda de modelos

Cuantificación

Principio: **reducir** el número de **bits necesarios** para codificar los parámetros del modelo.

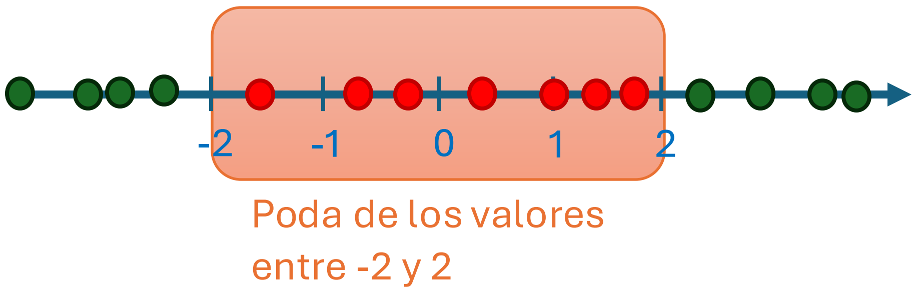


Ejemplo: Operador de redondeo:

- 1.4 ➔ 1
- 2.7 ➔ 3

Poda

Principio: **poner a cero** algunos parámetros del modelo.



Ejemplo: Poda de valores entre -2 y 2:

- 1.4 ➔ 0
- 2.7 ➔ 2.7
- 5.2 ➔ 5.2
- 0.7 ➔ 0

Etapas para la compresión de redes neuronales

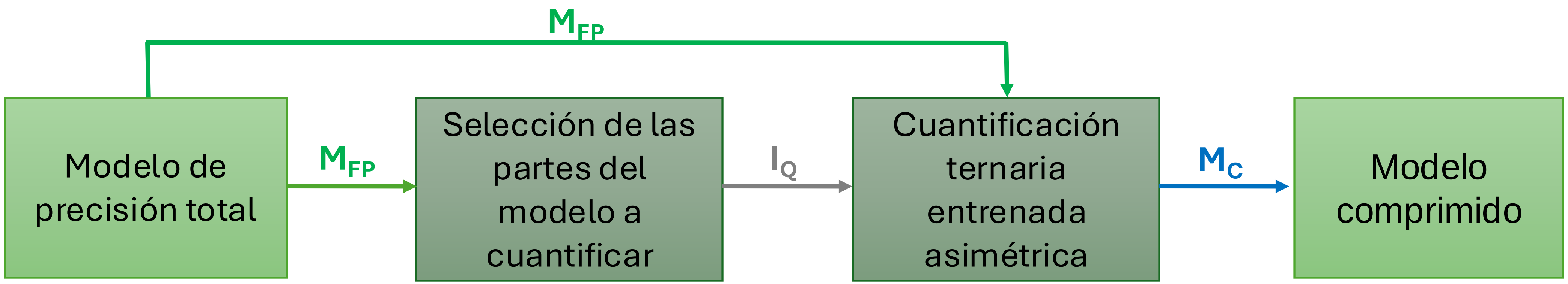
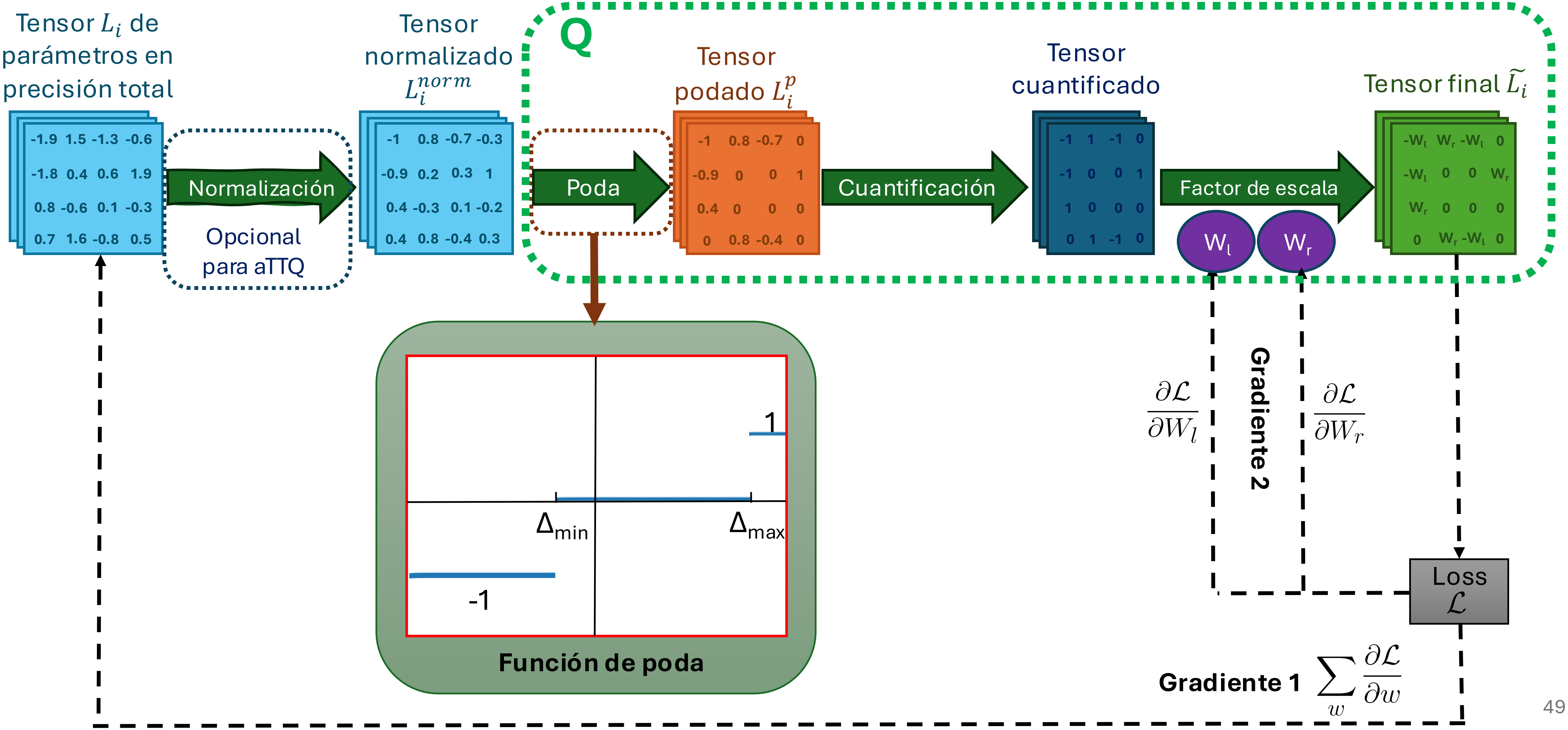


Figura – Etapas para la cuantificación ternaria asimétrica de redes neuronales

Asymmetric trained ternary quantization (aTTQ)



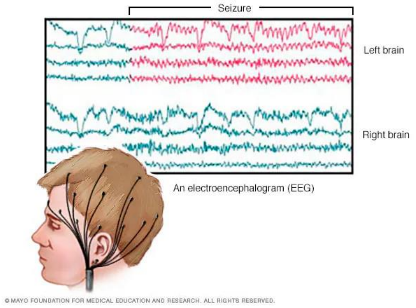
Experimento

Objetivo:

- Resaltar el interés de utilizar técnicas de compresión de modelos para ganar en energía y recursos de calculo

Bases de datos

HITS:
- Datos TCD.
- 1 545 **imágenes** y **señales** sin procesar manualmente anotados
- Tres clases.
- Frecuencia de muestreo: 4385 Hz.



ESR:
- Datos EEG.
- 11 500 muestras.
- Dos clases (crisis epiléptica y ausencia de crisis).
- Frecuencia de muestreo: 174 Hz.

Medidas de desempeño:

- Ahorro de energía(EC_S).
- Tasa de compresión(CR_G).
- ΔMCC, perdida de desempeño de clasificación.

Modelos:

- 2D CNN.
- 1D CNN-Transformer.

Experimento

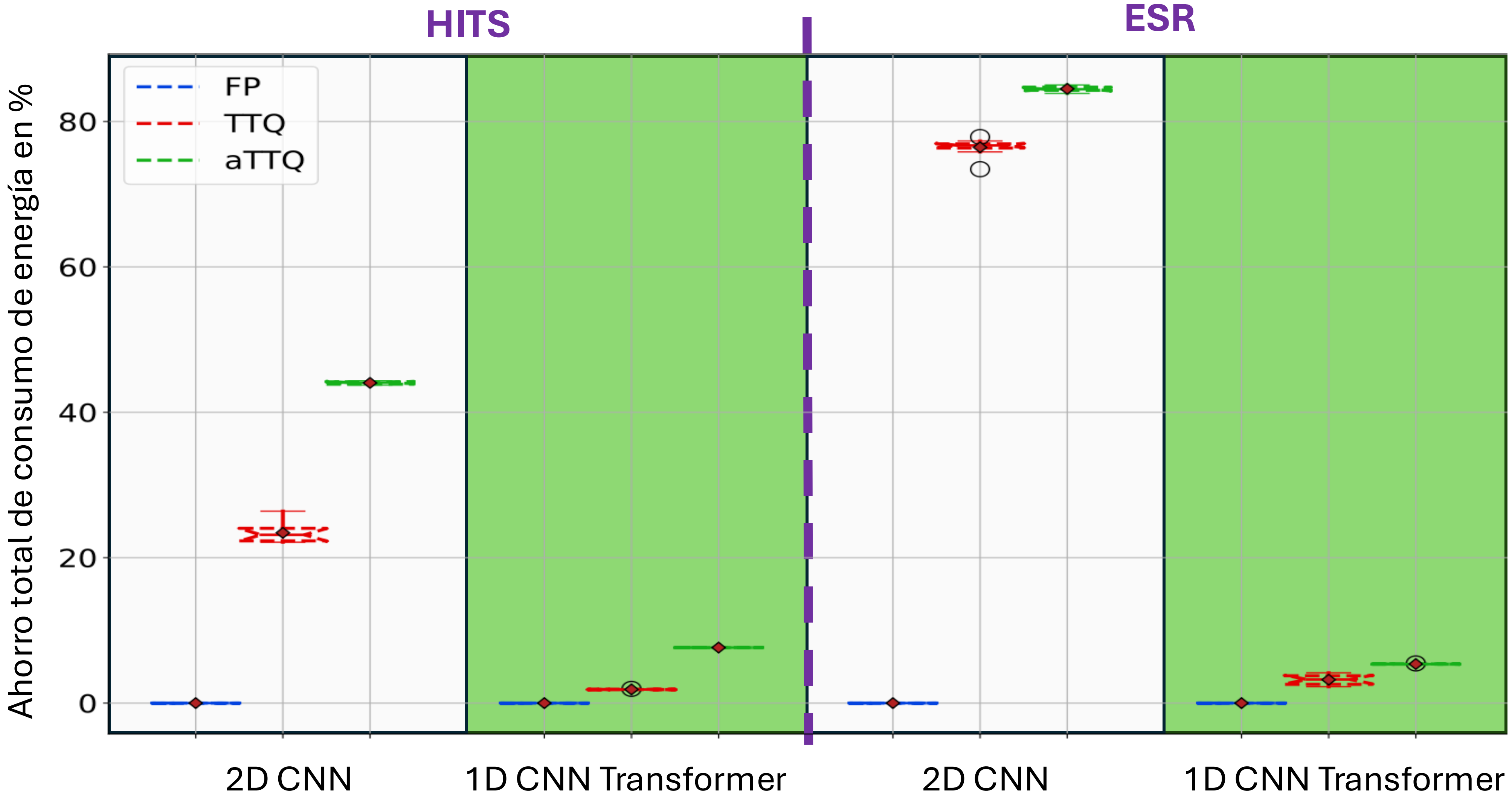


Figura – Comparación de modelos comprimidos con modelos en precisión total desde una perspectiva **energética.**

Experimento

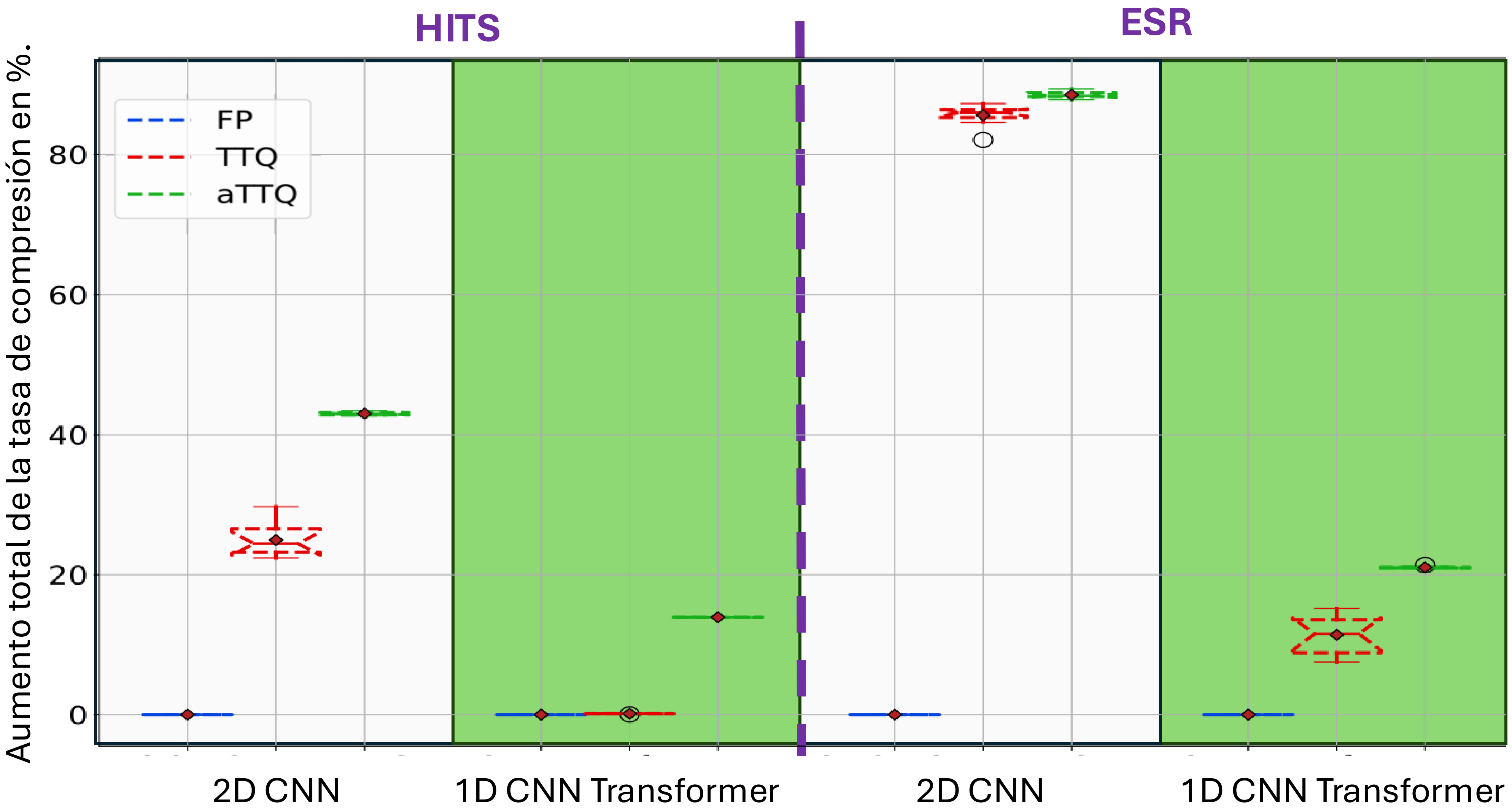


Figura – Comparación de modelos comprimidos con modelos en precisión total desde una perspectiva **compresión**.

Experimento

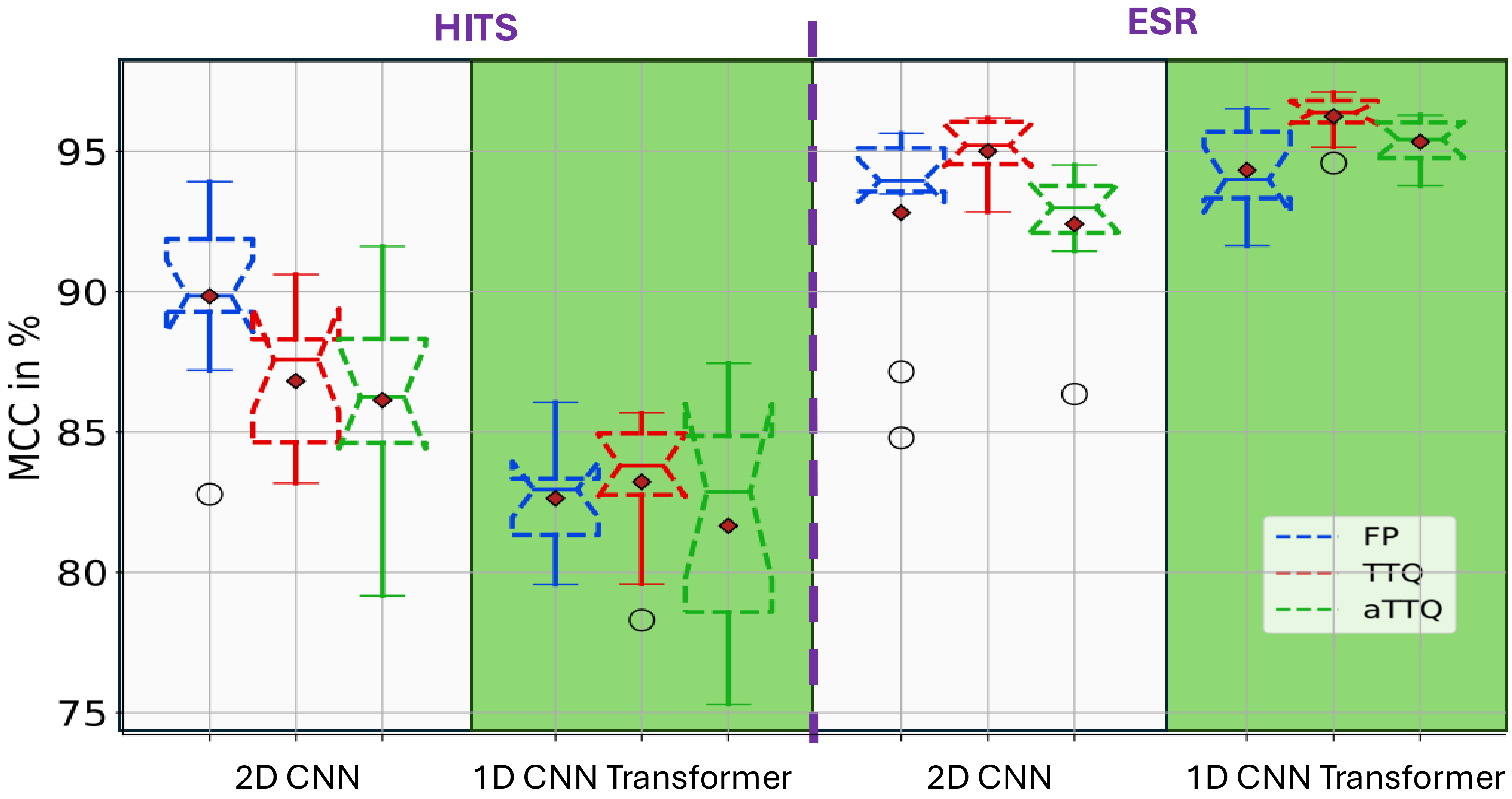


Figura – Comparación de modelos comprimidos con modelos en precisión total desde una perspectiva **de desempeño de clasificación**.

Estructura

I. Contexto
   a) Prevención de accidentes cerebrovasculares
   b) Otras aplicaciones de control médico
   c) Desafíos existentes

II. Introducción al aprendizaje automático
   a) Tipos de aprendizaje
   b) Reducción de dimensión
   c) Principio de entrenamiento

III. Inteligencia artificial para la medicina
   a) Anotación semiautomática de datos
   b) Modelos multi-representación
   c) Compresión de modelos

IV. **Conclusiones y perspectivas**

Posibles soluciones

Creación y anotación de conjuntos de datos	→	• Anotación de datos semi-supervisada* • Etiquetado flexible (anotación)*
Múltiples representaciones	→	• Diferentes modelos con diferentes entradas** • Modelo multi-representación
Modelos con gran demanda de recursos	→	• Modelos ligeros • Compresión de modelos*** • (Entrenamiento con etiquetas flexibles)

* Vindas et al. (IUS 2021), Vindas et al. (MEDIA 2022), Vindas et al. (IUS 2023)
** Vindas et al. (MLHC 2022), Vindas et al. (IABM 2023), Vindas et al. (EUSIPCO 2023) y Vindas et al. (Pattern Recognition 2023)
*** Vindas et al. (Neurocomputing 2024)

Perspectivas y aspectos para mejorar

Creación y anotación de bases de datos → **Aprendizaje activo**proponiendo a los expertos humanos las muestras más difíciles.

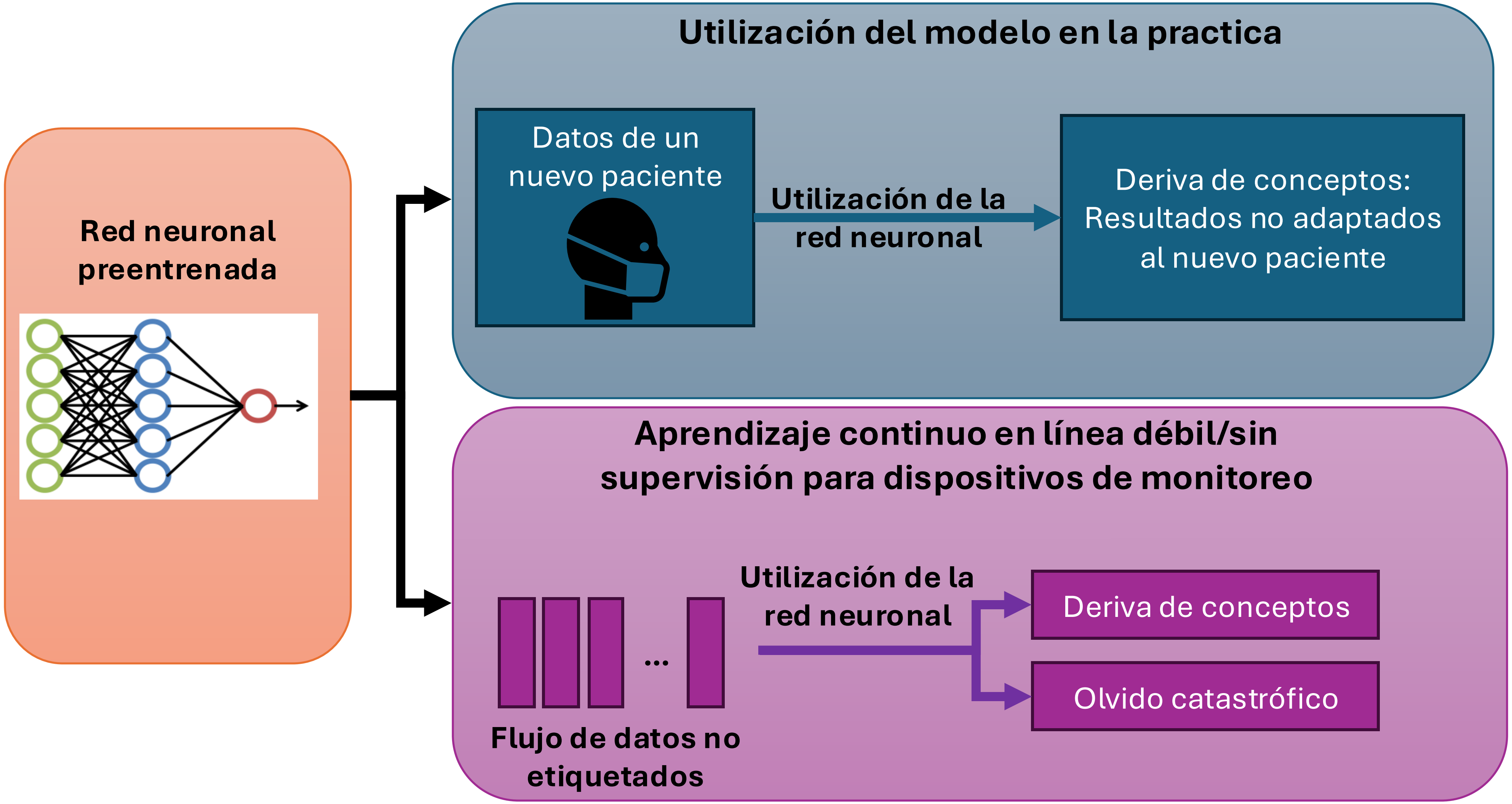
Utilizar **anotaciones flexibles** mediante **funciones de pérdida flexibles*** para capturar la incertidumbre del experto humano.

Múltiples representaciones → Utilizar **otros** tipos de **regularización** (aprendizaje contrastivo con supervisión débil, restricciones de enlace, ...).

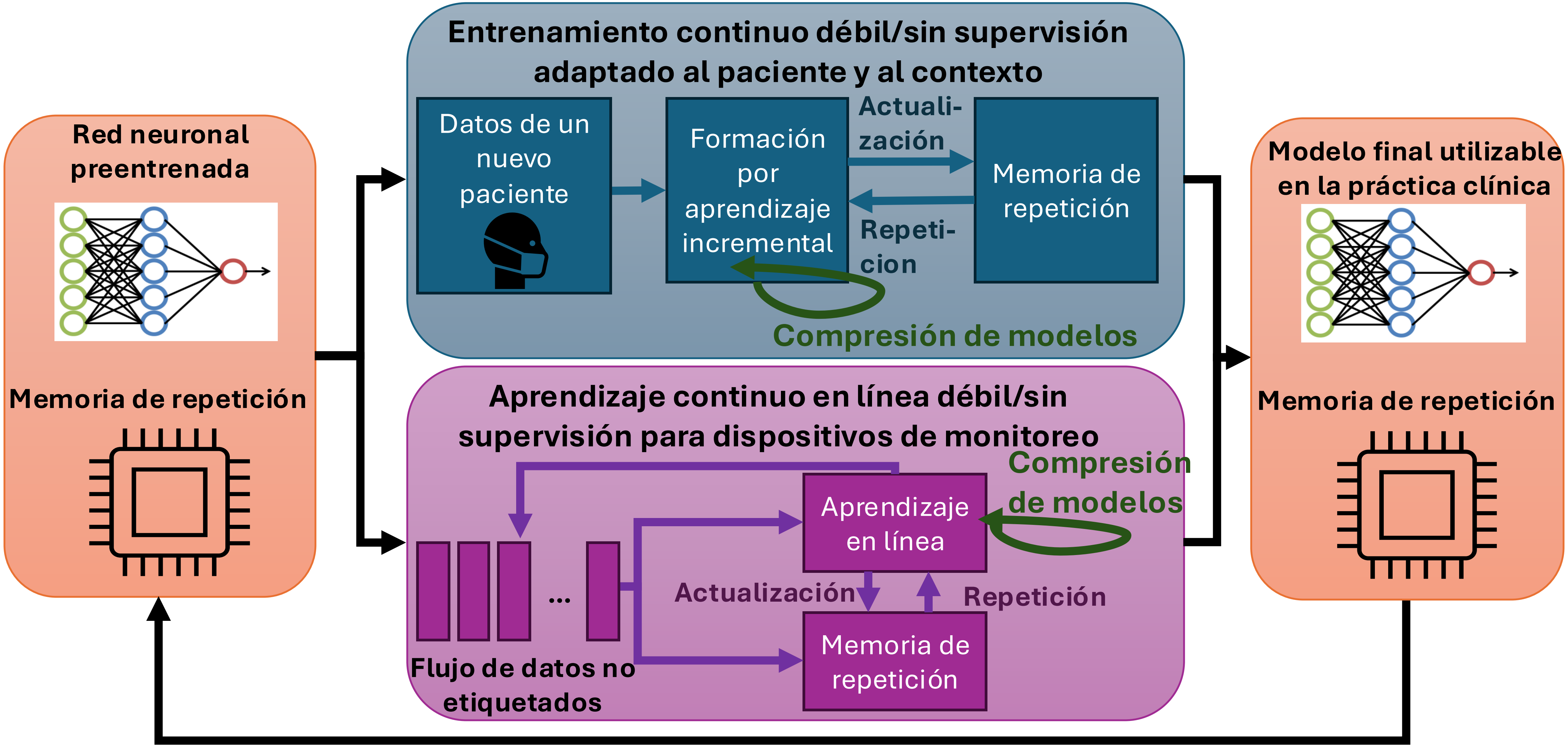
Modelos con gran demanda de recursos → **Función de poda** diferenciable, con **parámetros asimétricos** que se **pueden aprender**.

* Vindas, Y., Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2023. Soft-labels noise tolerant loss functions for transcranial Doppler ultrasound signal classification, in: 2023 IEEE International Ultrasonics Symposium (IUS)

Aprendizaje continuo



Aprendizaje continuo



Publicaciones

Revistas:

- **Vindas, Y.**, Guépie,´B.K., Almar, M., Roux, E., and Delachartre, P., 2022. Semi-automatic data annotation based on feature-space projection and local quality metrics: an application to cerebral emboli characterization, in **Medical Image Analysis**, page 102437, 2022. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2022.102437.
- **Vindas, Y.**, Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2023 Guided Deep Embedded Clustering regularization for multi-feature medical signal classification, **Pattern Recognition**, page 109812, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109812.
- **Vindas, Y.**, Guépie,´B.K., Almar, M., Roux, E., and Delachartre, P., 2023. Trainable pruned ternary quantization for medical signal classification models, **Neurocomputing**, page 128216, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2024.128216

Conferencias internacionales con actas:

- **Vindas, Y.**, Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2021. Semi-supervised annotation of transcranial Doppler ultrasound micro-embolic data, in: 2021 IEEE International Ultrasonics Symposium (**IUS**), pp. 1–4. doi:10.1109/IUS52206.2021.9593847.
- **Vindas, Y.**, Guépie,´B.K., Almar, M., Roux, E., and Delachartre, P., 2022. An hybrid CNN-Transformer model based on multi-feature extraction and attention fusion mechanism for cerebral emboli classification, in: **MLHC**. 05–06 Aug 2022, PMLR.
- **Vindas, Y.**, Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2023 Deep Embedded Clustering regularization for imbalanced cerebral emboli classification using transcranial Doppler ultrasound, in: European Signal Processing Conference (**EUSIPCO**) 04-08 Sep 2023
- **Vindas, Y.**, Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2023. Soft-labels noise tolerant loss functions for transcranial Doppler ultrasound signal classification, in: 2023 IEEE International Ultrasonics Symposium (**IUS**)

Conferencias nacionales sin actas :

- **Vindas, Y.**, Guépie,´B.K., Almar, M., Roux, E., and Delachartre, P., 2023. Classification multi-représentation d'emboles cérébraux à partir d'un dispositif de Doppler transcrânien. in:2023 Intelligence Artificielle en Imagerie Biomédicale (**IABM**).

Trabajos en curso:

- **Vindas, Y.**, Roux, E., Guépie,´B.K., Almar, M., Delachartre, P., 2023. An asymmetric heuristic for trained ternary quantization based on the weights' statistics: an application to medical signal classification. **(Minor Review)** to ***Pattern Recognition Letters***.

Gracias por su atención

References (1/3)

- Basic identification criteria of Doppler microembolic signals. Consensus Committee of the Ninth International Cerebral Hemodynamic Symposium. Stroke. 1995 Jun;26(6):1123. PMID: 7762033.
- Sombune, P., Phienphanich, P., Phuechpanpaisal, S., Muengtaweepongsa, S., Ruamthanthong, A. and Tantibundhit, C. (2017). Automated embolic signal detection using deep convolutional neural network, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp. 3365–3368.
- Guepie, B., Martin, M., Lacrosaz, V., Almar, M., Guibert, B. and Delachartre, P. (2018). Sequential emboli detection from ultrasound outpatient data, IEEE Journal of Biomedical and Health Informatics.
- Georgiadis, D., Grosset, D. G., Kelman, A., Faichney, A. and Lees, K. R. (1994). Prevalence and characteristics of intracranial microemboli signals in patients with different types of prosthetic cardiac valves., Stroke 25(3): 587–592.
- Aydin, N., & Markus, H. S. (2000). Optimization of processing parameters for the analysis and detection of embolic signals. European journal of ultrasound : official journal of the European Federation of Societies for Ultrasound in Medicine and Biology, 12(1), 69–79.
- N. Aydin, F. Marvasti and H. S. Markus, "Embolic Doppler ultrasound signal detection using discrete wavelet transform," in IEEE Transactions on Information Technology in Biomedicine, vol. 8, no. 2, pp. 182-190, June 2004.
- Marvasti, S., Gillies, D., Marvasti, F., & Markus, H. S. (2004). Online automated detection of cerebral embolic signals using a wavelet-based system. Ultrasound in medicine & biology, 30(5), 647–653.
- Markus, H. S. and Punter, M. (2005). Can Transcranial Doppler Discriminate Between Solid and Gaseous Microemboli?: Assessment of a Dual-Frequency Transducer System, Stroke 36(8): 1731–1734.
- Chung, E., Fan, L., Degg, C., & Evans, D. H. (2005). Detection of Doppler embolic signals: psychoacoustic considerations. Ultrasound in medicine & biology, 31(9), 1177–1184.
- Gençer, M., Bilgin, G., & Aydın, N. (2013). Embolic Doppler ultrasound signal detection via fractional Fourier transform. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, *2013*, 3050–3053.
- Serbes, G. and Aydin, N. (2014). Denoising performance of modified dual-tree complex wavelet transform for processing quadrature embolic doppler signals, Medical & Biological Engineering & Computing 52(1): 29–43.
- Karahoca, A. and Tunga, M. A. (2015). A polynomial based algorithm for detection of embolism, 19(1): 167–177.
- Imaduddin, S. M., LaRovere, K. L., Kussman, B. D. and Heldt, T. (2019). A time-frequency approach for cerebral embolic load monitoring, IEEE Transactions on Biomedical Engineering 67(4): 1007–1018.
- Darbellay GA, Duff R, Vesin J-M, et al. Solid or Gaseous Circulating Brain Emboli: Are They Separable by Transcranial Ultrasound? *Journal of Cerebral Blood Flow & Metabolism*. 2004;24(8):860-868.
- Keunen, R. W. M., Hoogenboezem, R., Wijnands, R., Van den Hengel, A. C. M. and Ackerstaff, R. G. A. (2008). Introduction of an embolus detection system based on analysis of the transcranial Doppler audio-signal, Journal of Medical Engineering & Technology 32(4): 296–304.
- A. Karahoca, T. Kucur and N. Aydin, "Data Mining Usage in Emboli Detection," 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security (BLISS 2007), Edinburgh, UK, 2007, pp. 159-162.
- Guépié, B. K., Sciolla, B., Millioz, F., Almar, M. and Delachartre, P. (2017). Discrimination between emboli and artifacts for outpatient transcranial doppler ultrasound data, Medical & Biological Engineering & Computing 55(10): 1787–1797. Number: 10.
- Chen, Yijiao & Yuanyuan, Wang. (2008). Doppler embolic signal detection using the adaptive wavelet packet basis and neurofuzzy classification. Pattern Recognition Letters.
- Sombune, P., Phienphanich, P., Muengtaweepongsa, S., Ruamthanthong, A. and Tantibundhit, C. (2016). Automated embolic signal detection using adaptive gain control and classification using anfis, 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 3825–3828.
- Tafsast, A., Ferroudji, K., Hadjili, M. L., Bouakaz, A. and Benoudjit, N. (2018). Automatic microemboli characterization using convolutional neural networks and radio frequency sig- nals, 2018 International Conference on Communications and Electrical Engineering (ICCEE), IEEE, pp. 1–4.
- Benato, B. C., Gomes, J. F., Telea, A. C. and Falcão, A. X. (2021). Semi-automatic data annotation guided by feature space projection, Pattern Recognition 109: 107612.
- Lueks, W., Mokbel, B., Biehl, M. and Hammer, B. (2011). How to evaluate dimensionality reduction? - improving the co-ranking matrix, arXiv:1110.3917 [cs] .
- Zhilu Zhang, Mert R. Sabuncu: Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. NeurIPS 2018: 8792-8802.
- Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, eds.Semi-supervised learning. Adaptive computation and machine learning. OCLC: ocm64898359. Cambridge, Mass: MIT Press, 2006. 508 pp.isbn: 978-0-262-03358-9.
- Amorim, W. P., Falcão, A. and Carvalho, M. H. (2014). Semi-supervised pattern classification using optimum-path forest, 2014 27th SIBGRAPI Conference on Graphics, Patterns and Images pp. 111–118.
- Melacci, Stefano & Belkin, Mikhail. (2009). Laplacian Support Vector Machines Trained in the Primal. Journal of Machine Learning Research.
- Pu, J., Panagakis, Y. and Pantic, M. (2021). Learning separable time-frequency filterbanks for audio classification, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3000–3004
- Lee, J., Park, J., Kim, K. and Nam, J. (2017). Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms, 14th Sound Music Computing Conference.
- Park, H. and Yoo, C. D. (2020). Cnn-based learnable gammatone filterbank and equal-loudness normalization for environmental sound classification, IEEE Signal Processing Letters 27: 411–415.
- Sharan, R., Xiong, H. and Berkovsky, S. (2021). Benchmarking audio signal representation techniques for classification with convolutional neural networks, Sensors 21: 3434.
- Yeh, C.-F., Mahadeokar, J., Kalgaonkar, K., Wang, Y., Le, D., Jain, M., Schubert, K., Fuegen, C. and Seltzer, M. L. (2019). Transformer-transducer: End-to-end speech recognition with self-attention, ArXiv abs/1910.12977.
- Natarajan, A., Chang, Y., Mariani, S., Rahman, A., Boverman, G., Vij, S. and Rubin, J. (2020). A wide and deep transformer neural network for 12-lead ecg classification, 2020 Computing in Cardiology, pp. 1–4. xiii.
- Okawa, M., Saito, T., Sawada, N. and Nishizaki, H. (2019). Audio classification of bit- representation waveform, INTERSPEECH, pp. 2553–2557.
- Nishizaki, H. and Makino, K. (2019). Signal classification using deep learning, pp. 1–4.
- Karita, S., Wang, X., Watanabe, S., Yoshimura, T., Zhang, W., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. and Yamamoto, R. (2019). A comparative study on transformer vs RNN in speech applications, 2019 IEEE Automatic Speech Recognition and Understanding Workshop.
- Boes, W. and Van hamme, H. (2019). Audiovisual transformer architectures for large-scale classification and synchronization of weakly labeled audio events, Proceedings of the 27th ACM International Conference on Multimedia, MM '19, Association for Computing Machinery, New York, NY, USA, p. 1961–1969.
- Mohamed, Abdelrahman & Okhonko, Dmytro & Zettlemoyer, Luke. (2019). Transformers with convolutional context for ASR.
- kbari, H., Yuan, L., Qian, R., Chuang, W.-H., Chang, S.-F., Cui, Y. and Gong, B. (2021). Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, Advances in Neural Information Processing Systems 34. 50, 54.
- Ding, Y., Jia, M., Miao, Q. and Cao, Y. (2022). A novel time–frequency transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings, Mechanical Systems and Signal Processing 168: 108616.

References (1/3)

- Che, C., Zhang, P., Zhu, M., Qu, Y. and Jin, B. (2021). Constrained transformer network for ecg signal processing and arrhythmia classification, BMC Medical Informatics and Decision Making 21.
- Gong, Y., Chung, Y.-A. and Glass, J. (2021). AST: Audio Spectrogram Transformer, Proc. Interspeech 2021, pp. 571–575.
- Xie, J., Girshick, R. and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis, Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, JMLR.org, p. 478–487.
- Zhu, Y. and Jiang, Y. (2020). Optimization of face recognition algorithm based on deep learning multi feature fusion driven by big data, Image and Vision Computing 104: 104023.
- Ahmad, Z., Tabassum, A., Guan, L. and Khan, N. M. (2021). Ecg heartbeat classification using multimodal fusion, IEEE Access.
- Yao, T., Gao, F., Zhang, Q. and Ma, Y. (2021). Multi-feature gait recognition with dnn based on semg signals, Mathematical Biosciences and Engineering 18: 3521–3542.
- Wang, L., Zhang, J., Liu, P., Choo, K.-K. R. and Huang, F. (2017). Spectral–spatial multi-feature- based deep learning for hyperspectral remote sensing image classification, Soft Computing 21.
- Feng, X., Feng, Q., Li, S., Hou, X. and Liu, S. (2020). A deep-learning-based oil-well-testing stage interpretation model integrating multi-feature extraction methods, Energies 13(8).
- Kim, J.-G. and Lee, B. (2019). Appliance classification by power signal analysis based on multi- feature combination multi-layer lstm, Energies 12(14).
- Chen, X., Cheng, Z., Wang, S., Lu, G., Xv, G., Liu, Q. and Zhu, X. (2021). Atrial fibrillation detection based on multi-feature extraction and convolutional neural network for processing ecg signals, Computer Methods and Programs in Biomedicine 202: 106009.
- Abdi, Asad & Shamsuddin, Siti Mariyam & Piran, Jalil. (2019). Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion. Information Processing & Management. 56. 1245-1259.
- Tongxue Zhou, Su Ruan, Pierre Vera, Stéphane Canu, "A Tri-attention Fusion Guided Multi-modal Segmentation Network Pattern Recognition", Elsevier, Pattern Recognition, Volume 124, 108417, April 2022.
- Tongxue Zhou, Stéphane Canu, Su Ruan, "Fusion based on attention mechanism and context constrain for multi-modal brain tumor segmentation", Elsevier, Computerized Medical Imaging and Graphics, Volume 86, 101811. December 2020.
- Mao, S., Li, Y., Ma, Y., Zhang, B., Zhou, J. and Wang, K. (2020). Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi- feature fusion, Computers and Electronics in Agriculture 170.
- Tiong, L., Kim, S. T. and Ro, Y. (2019). Implementation of multimodal biometric recognition via multi-feature deep learning networks and feature fusion, Multimedia Tools and Applications 78.
- Liu, Z.-M. (2021). Multi-feature fusion for specific emitter identification via deep ensemble learning, Digital Signal Processing 110: 102939.
- Jin, J., Yang, S., Zhao, B., Luo, L. and Woo, W. L. (2020). Attention-block deep learning based features fusion in wearable social sensor for mental wellbeing evaluations, IEEE Access 8: 1–1.
- Horowitz, M. (2014). 1.1 computing's energy problem (and what we can do about it), 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 10–14.
- Molka, D., Hackenberg, D., Schöne, R. and Müller, M. S. (2010). Characterizing the energy consumption of data transfers and arithmetic operations on x8664 processors, International Conference on Green Computing, pp. 123–133.
- Gong, Y., Liu, L., Yang, M. and Bourdev, L. D. (2014). Compressing deep convolutional networks using vector quantization.
- Kim, Hyeji & Khan, Muhammad Umar Karim & Kyung, Chong-Min. (2019). Efficient Neural Network Compression.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R. (2020). ALBERT: A lite BERT for self-supervised learning of language representations, 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.
- Zhang, D., Yang, J., Ye, D. and Hua, G. (2018). Lq-nets: Learned quantization for highly accurate and compact deep neural networks, Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII, Springer-Verlag, Berlin, Heidelberg, p. 373–390.
- Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J. and Hua, X.-s. (2019). Quantization networks, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- Rastegari, M., Ordonez, V., Redmon, J. and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks, in B. Leibe, J. Matas, N. Sebe and M. Welling (eds), Computer Vision – ECCV 2016, Springer International Publishing, Cham, pp. 525–542.
- Prato, G., Charlaix, E. and Rezagholizadeh, M. (2020). Fully quantized transformer for machine translation, Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, pp. 1–14.
- Zhu, C., Han, S., Mao, H. and Dally, W. J. (2017). Trained ternary quantization, International Conference on Learning Representations.
- Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T. and Kwak, N. (2020). Lsq+: Improving low-bit quantization through learnable offsets and better initialization, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
- Dong, Z., Yao, Z., Arfeen, D., Gholami, A., Mahoney, M. W. and Keutzer, K. (2020). Hawq-v2: Hessian aware trace-weighted quantization of neural networks, in H. Larochelle, M. Ran- zato, R. Hadsell, M. Balcan and H. Lin (eds), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., pp. 18518–18529.
- Xu, Y., Wang, Y., Zhou, A., Lin, W. and Xiong, H. (2018). Deep neural network compres- sion with single and multiple level quantization, Proceedings of the AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18, AAAI Press.
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Kim, S., Gholami, A., Yao, Z., Mahoney, M. W. and Keutzer, K. (2021). I-bert: Integer-only bert quantization, International Conference on Machine Learning.
- Zhou, S., Ni, Z., Zhou, X., Wen, H., Wu, Y. and Zou, Y. (2016). Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients, CoRR abs/1606.06160.
- Han, S., Mao, H. and Dally, W. J. (2016). Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding, in Y. Bengio and Y. LeCun (eds), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico.
- Li, F. and Liu, B. (2016). Ternary weight networks, ArXiv abs/1605.04711.
- Park, M. S., Xu, X. and Brick, C. (2018). Squantizer: Simultaneous learning for both sparse and low-precision neural networks, CoRR abs/1812.08301.
- Ullrich, K., Meeds, E. and Welling, M. (2017). Soft weight-sharing for neural network compres- sion, International Conference on Learning Representations.
- Tung, F. and Mori, G. (2020). Deep neural network compression by in-parallel pruning- quantization, IEEE Transactions on Pattern Analysis and Machine Intelligence 42(3): 568–579.
- Ji, T., Jain, S., Ferdman, M., Milder, P., Schwartz, H. A. and Balasubramanian, N. (2021). On the distribution, sparsity, and inference-time quantization of attention values in transformers, ArXiv abs/2106.01335.

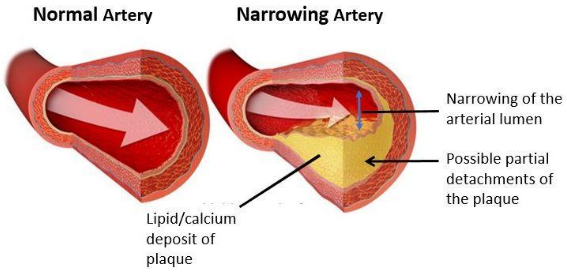
References (1/3)

- Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N. and Peste, A. (2022). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, J. Mach. Learn.
- Han, S., Pool, J., Tran, J. and Dally, W. (2015). Learning both weights and connections for efficient neural network, Advances in neural information processing systems 28.
- Hassibi, B., Stork, D. and Wolff, G. (1993). Optimal brain surgeon and general network pruning, IEEE International Conference on Neural Networks, pp. 293–299 vol.1.
- Mariet, Z. and Sra, S. (2016). Diversity networks: Neural network compression using determinantal point processes, International Conference on Learning Representations (ICLR).
- Zhu, M. and Gupta, S. (2018). To prune, or not to prune: Exploring the efficacy of pruning for model compression, 6th International Conference on Learning Representations, ICLR 2018.
- Manessi, F., Rozza, A., Bianco, S., Napoletano, P. and Schettini, R. (2017). Automated pruning for deep neural network compression.
- Luo, J.-H., Wu, J. and Lin, W. (2017). Thinet: A filter level pruning method for deep neural network compression, pp. 5068–5076.
- He, Y., Lin, J., Liu, Z., Wang, H., Li, L.-J. and Han, S. (2018). Amc: Automl for model compression and acceleration on mobile devices, in V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (eds), Computer Vision – ECCV 2018, Springer International Publishing, Cham, pp. 815–832.
- Xu, K., Zhang, D., An, J., Liu, L., Liu, L. and Wang, D. (2021). Genexp: Multi-objective pruning for deep neural network based on genetic algorithm, Neurocomputing 451: 81–94.
- Bai, H., Zhang, W., Hou, L., Shang, L., Jin, J., Jiang, X., Liu, Q., Lyu, M. and King, I. (2021). BinaryBERT: Pushing the limit of BERT quantization, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, pp. 4334–4348.
- Sun, S., Cheng, Y., Gan, Z. and Liu, J. (2019). Patient knowledge distillation for bert model compression, Conference on Empirical Methods in Natural Language Processing.
- Zhang, W., Hou, L., Yin, Y., Shang, L., Chen, X., Jiang, X. and Liu, Q. (2020). Ternarybert: Distillation-aware ultra-low bit bert, Conference on Empirical Methods in Natural Language Processing.
- Polino, A., Pascanu, R. and Alistarh, D. (2018). Model compression via distillation and quanti- zation, ArXiv e-prints .
- Kanjilal, P., Dey, P. and Banerjee, D. (1993). Reduced-size neural networks through singular value decomposition and subset selection, Electronics Letters 29: 1516–1518.
- Yao, Z., Dong, Z., Zheng, Z., Gholami, A., Yu, J., Tan, E., Wang, L., Huang, Q., Wang, Y., Mahoney, M. and Keutzer, K. (2021). Hawq-v3: Dyadic neural network quantization, in M. Meila and T. Zhang (eds), Proceedings of the 38th International Conference on Machine Learning, Vol. 139 of Proceedings of Machine Learning Research, PMLR, pp. 11875–11886.
- Hinton, G. E., Vinyals, O. and Dean, J. (2015). Distilling the knowledge in a neural network, ArXiv abs/1503.02531.
- Bosio, A., O'Connor, I., Traiola, M., Echavarria, J., Teich, J., Hanif, M. A., Shafique, M., Ham- dioui, S., Deveautour, B., Girard, P., Virazel, A. and Bertels, K. (2021). Emerging computing devices: Challenges and opportunities for test and reliability , 26th IEEE European Test Sym-posium, ETS 2021, Bruges, Belgium, May 24-28, 2021, IEEE, pp. 1–10.
- Yu, F. and Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions, in Y. Bengio and Y. LeCun (eds), 4th International Conference on Learning Representations, ICLR.
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications.
- He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
- Gao, K., Zhang, Q. and Wang, H. (2019). A lightweight residual-inception convolutional neural network, Journal of Physics: Conference Series 1237: 032058.
- Elsken, T., Metzen, J. H. and Hutter, F. (2019). Neural architecture search: A survey, J. Mach. Learn. Res. 20(1): 1997–2017.

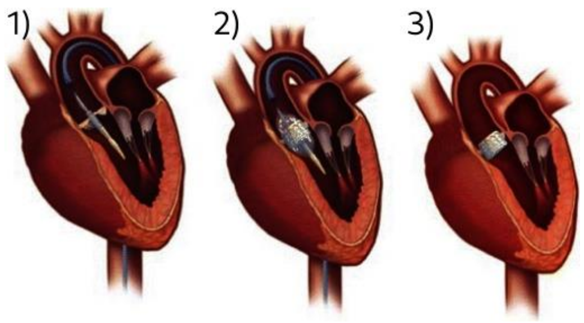
Backslides

Context

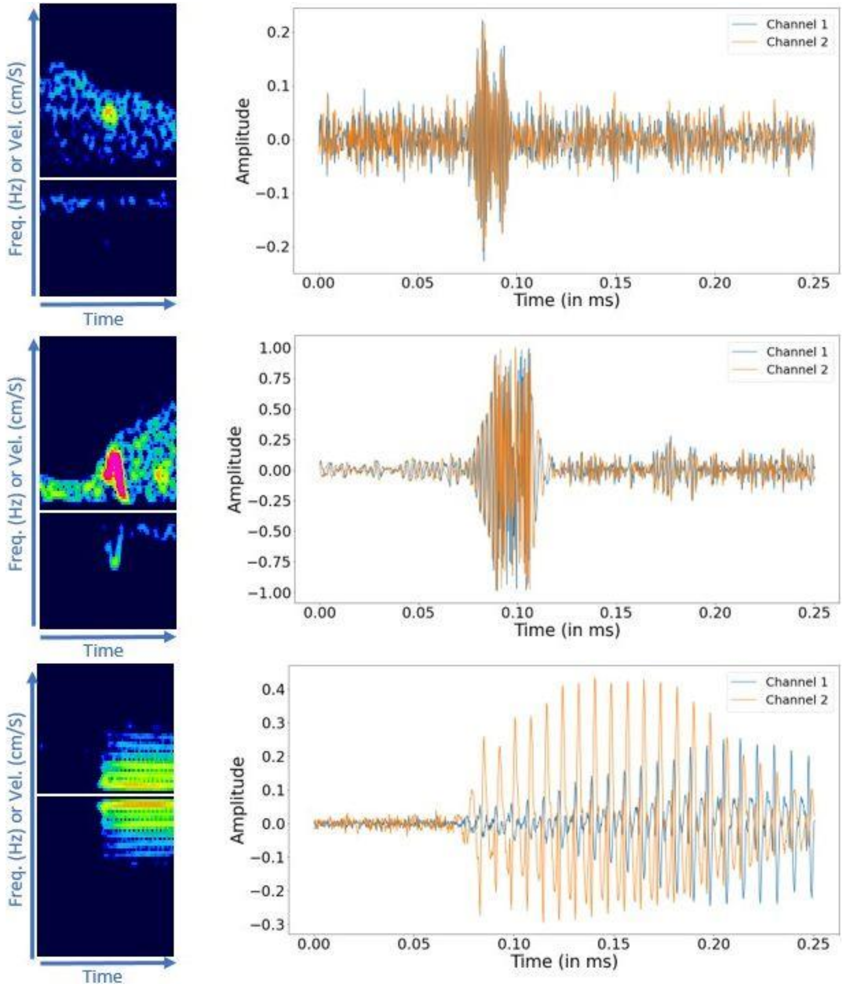
WHY ?

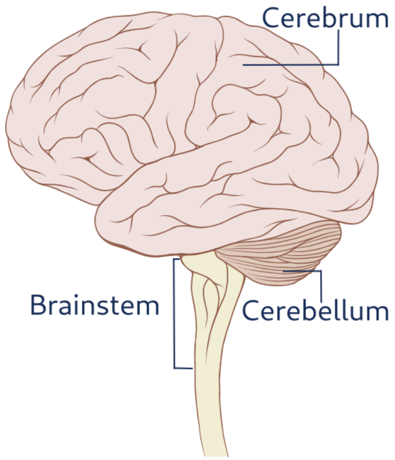


Atherosclerosis



Transcatheter aortic valve replacement



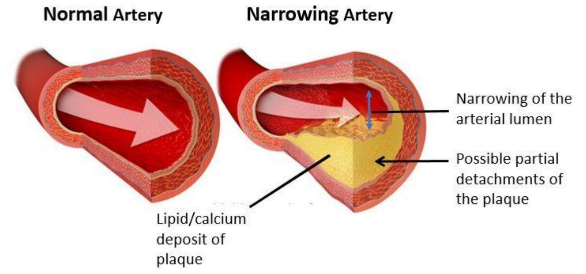


Different Sources

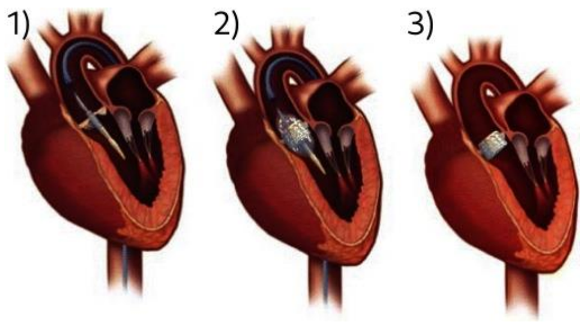
Different Types

Different Consequenc

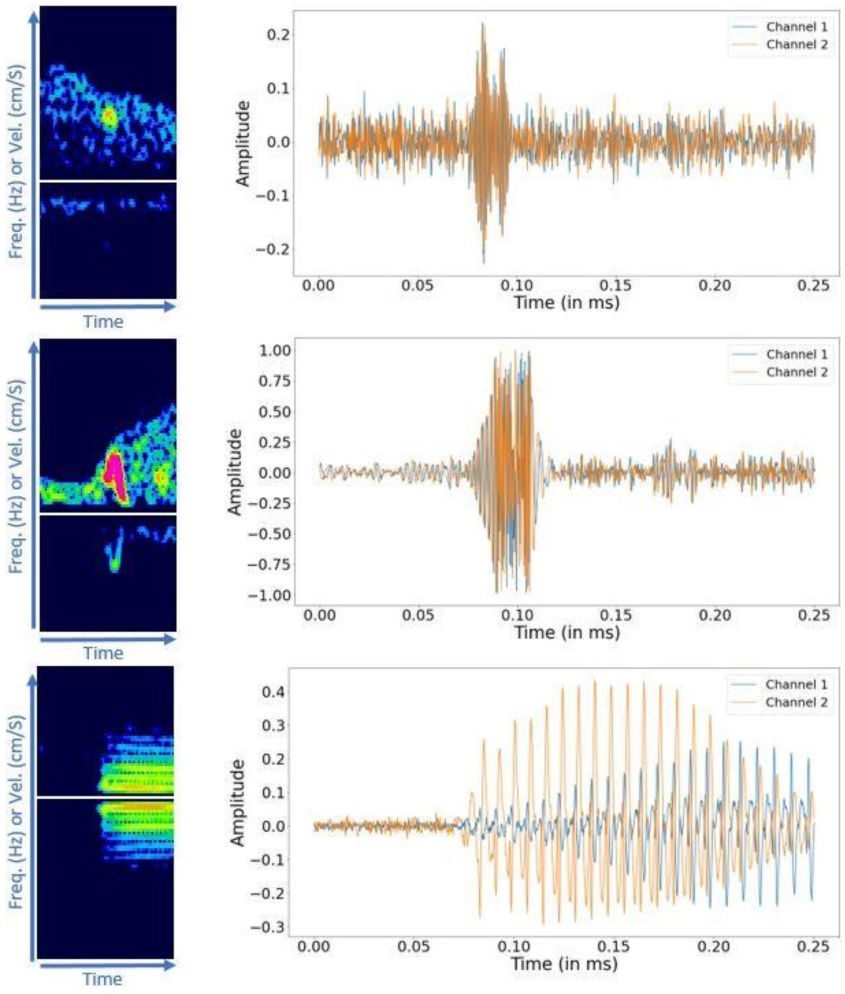
WHY ?

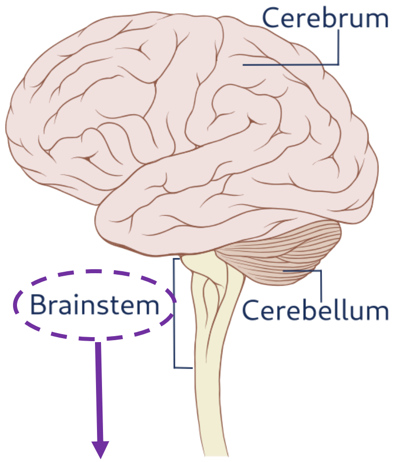


Atherosclerosis



Transcatheter aortic valve replacement





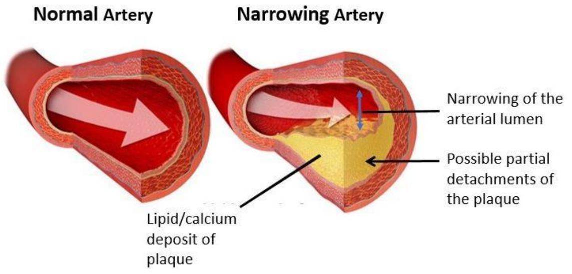
- Temperature regulation.
- Heart rate and blood pressure.
- Vision.
- Balance and coordination.
- Swallowing.
- Coma or death.

Different Sources

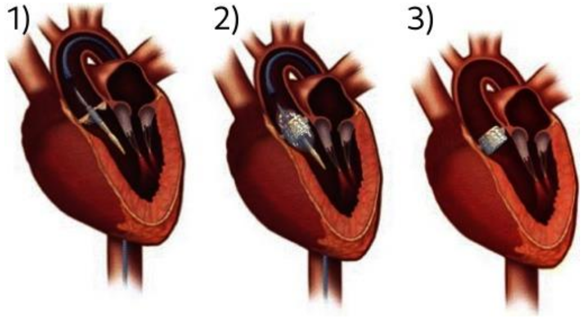
Different Types

Different Consequenc

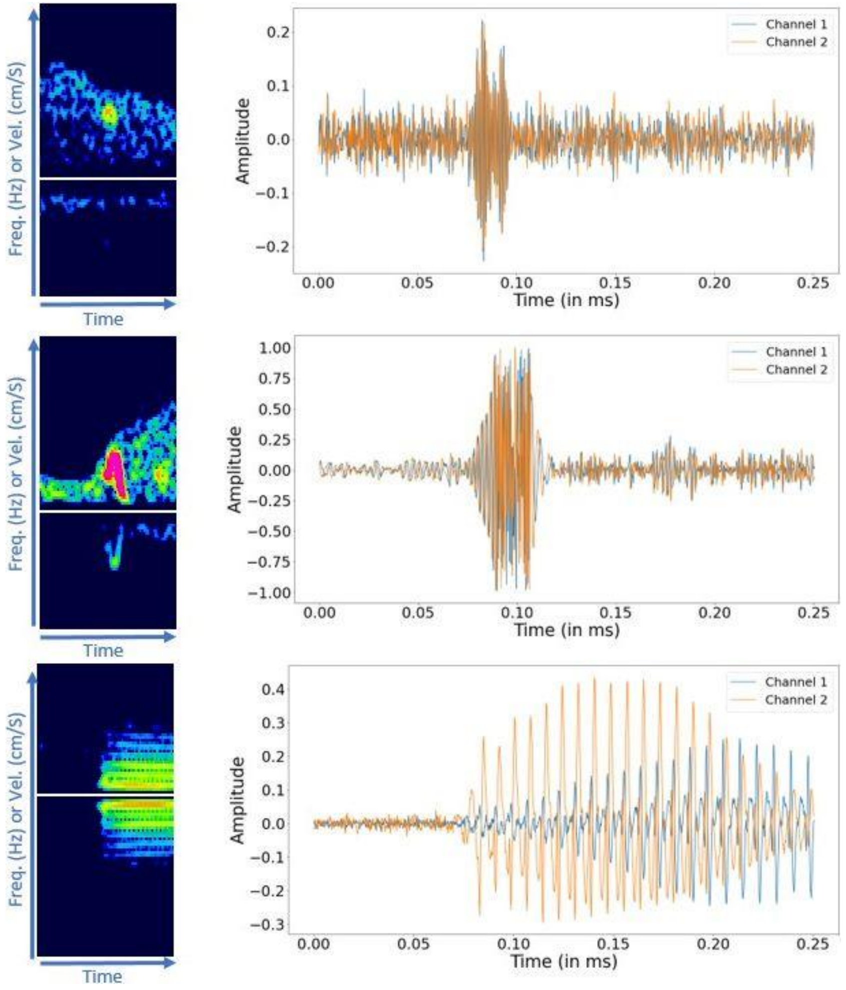
WHY ?

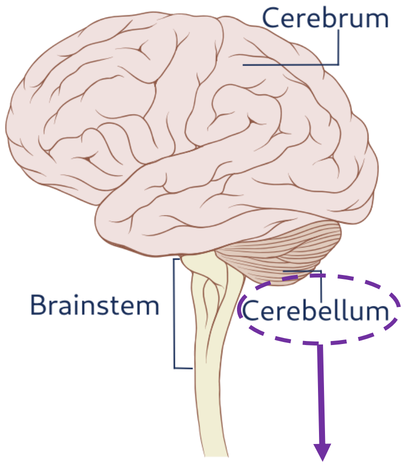


Atherosclerosis



Transcatheter aortic valve replacement





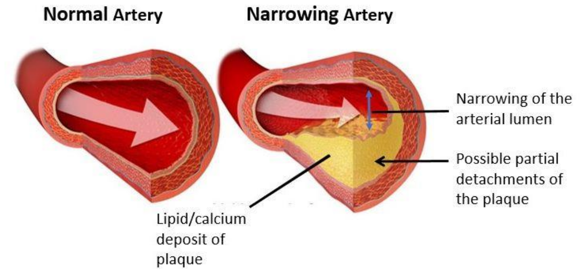
- Balance and posture.
- Muscle movements.
- Headache.
- Nausea.

Different Sources

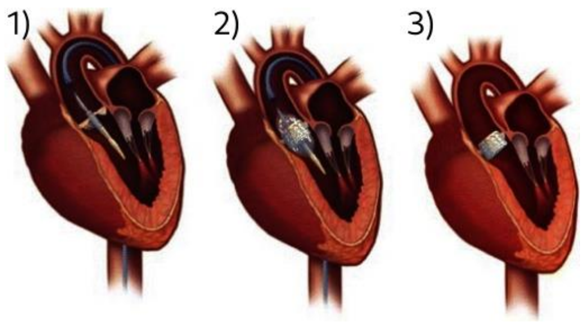
Different Types

Different Consequenc

WHY ?

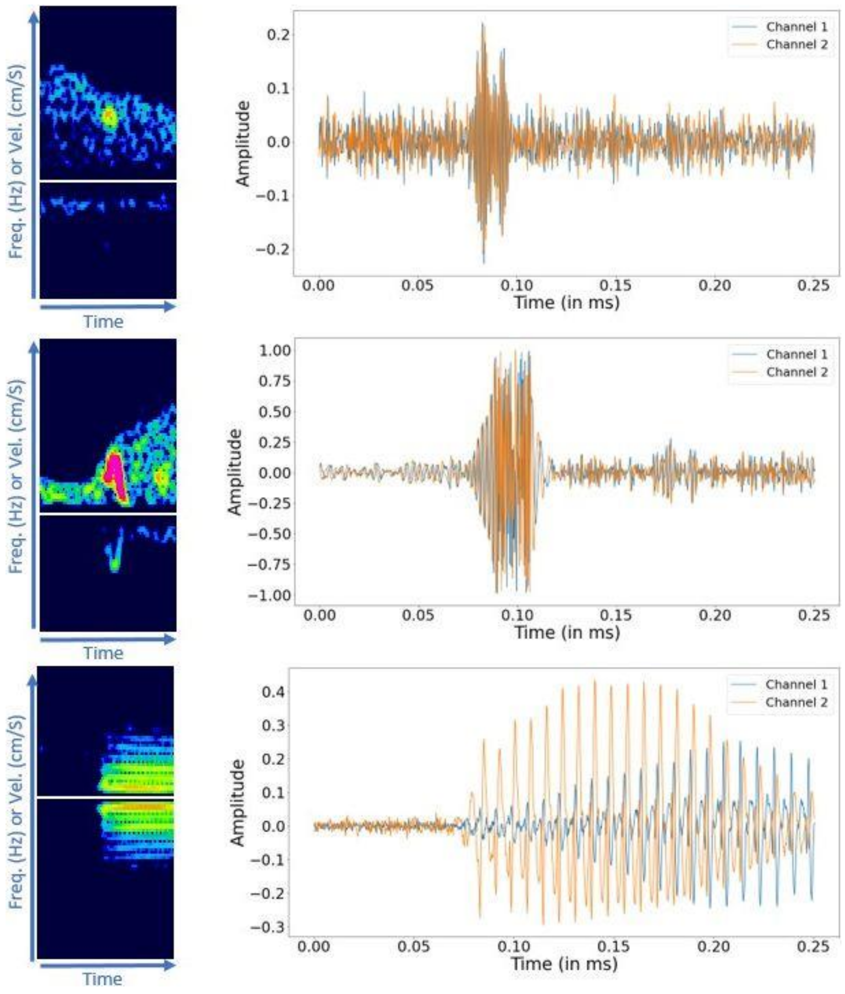


Atherosclerosis



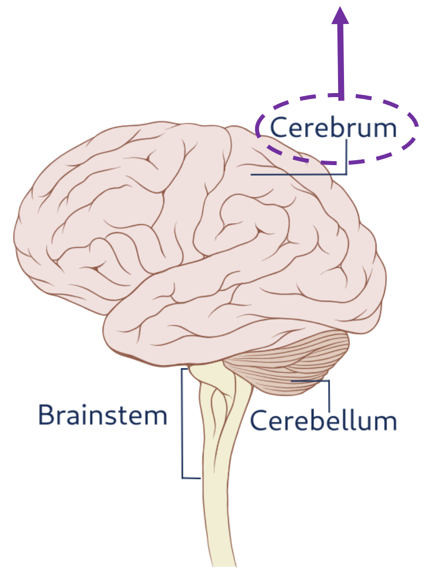
Transcatheter aortic valve replacement

Different Sources



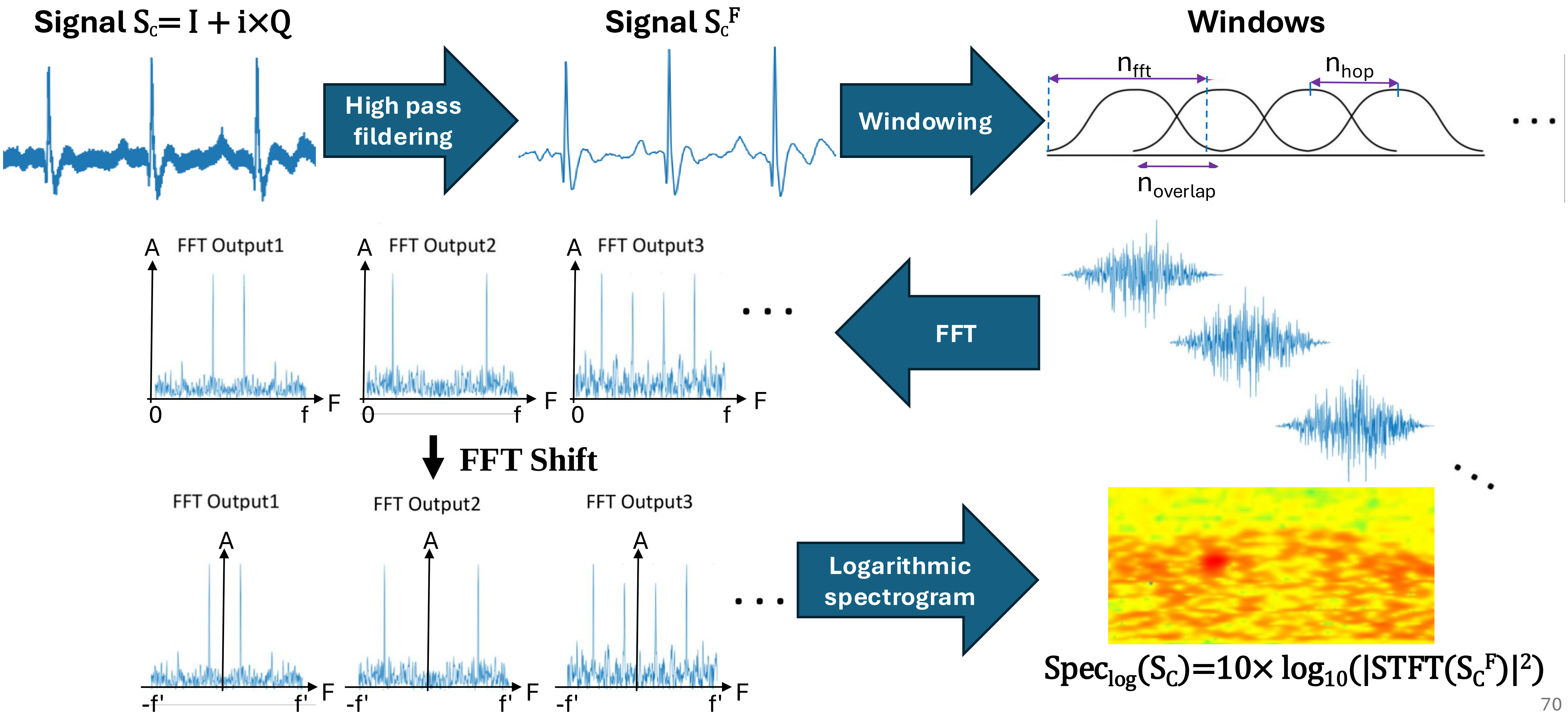
Different Types

- Memory.
- Reasoning.
- Paralysis.
- Vision and speech.
- Behavior.

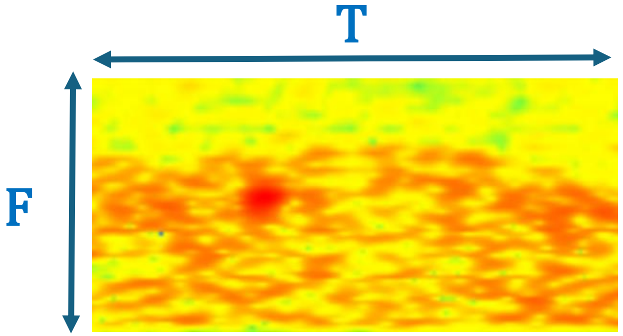


Different Consequenc

Time-frequency representations

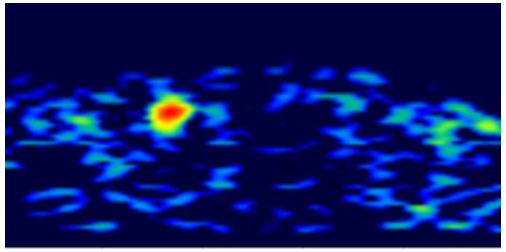


Time-frequency representations



$$Spec_{log}(S_C)=10\times \log_{10}(|STFT(S_C^F)|^2)$$

Filtering



$$Spec_{final}(S_C)=Filter(Spec_{log}(S_C))$$

$$\forall i\in[1,F],\ j\in[1,F],\ Spec_{final}(S_C) = \begin{cases} min_{dB} \text{ if } Spec_{log}(S_C)[i,j] < min_{dB}=\mu_{Spec}+a\times\sigma_{Spec} \\ Spec_{log}(S_C)[i,j] \text{ if } Spec_{log}(S_C)[i,j] \in [min_{dB}, max_{dB}] \\ max_{dB} \text{ if } Spec_{log}(S_C)[i,j] > max_{dB}=\mu_{Spec}+b\times\sigma_{Spec} \end{cases}$$

Emboli detection

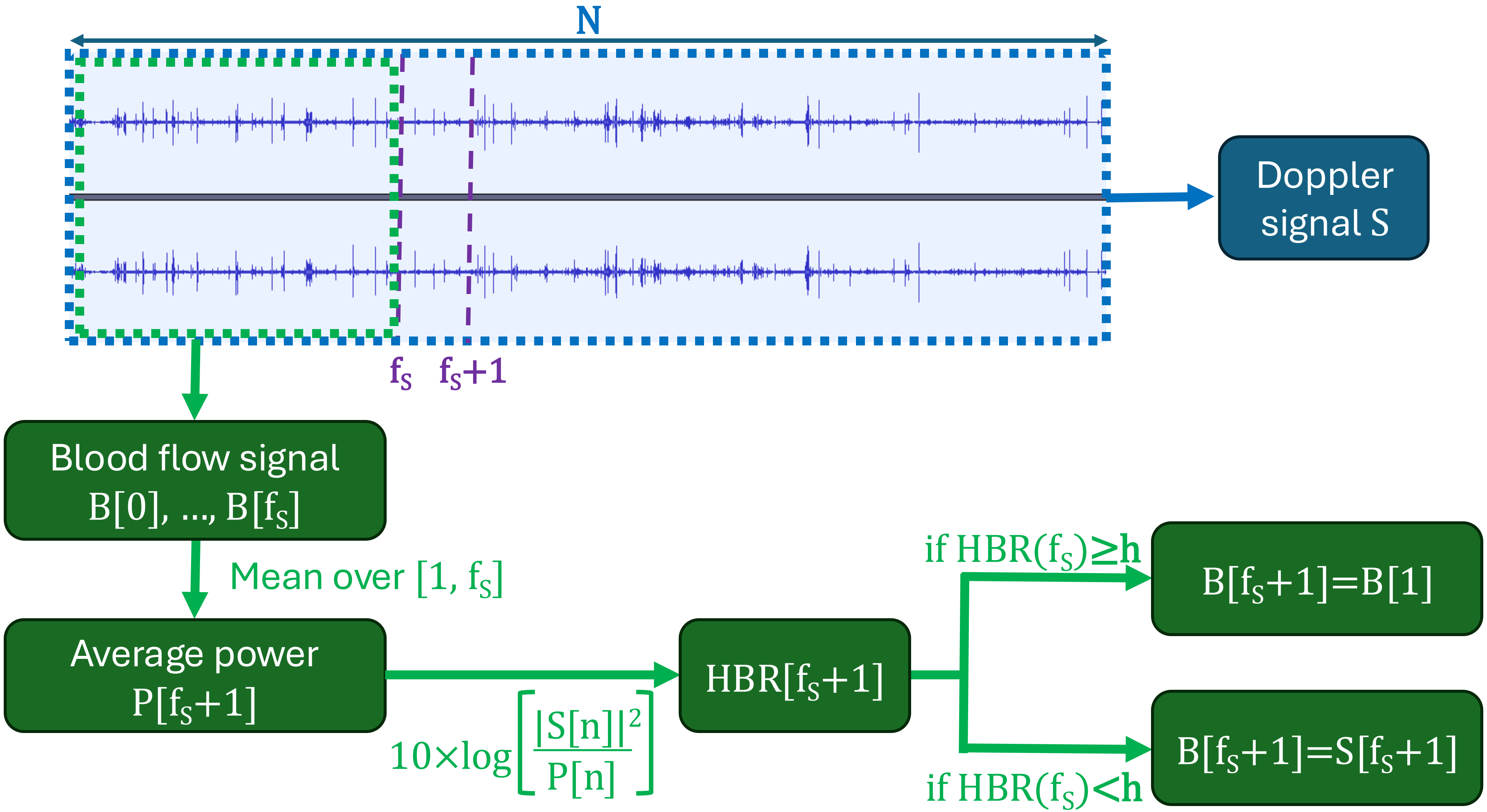


Figure – HBR computation **initialization** from Guépié et al. (2019)

Emboli detection

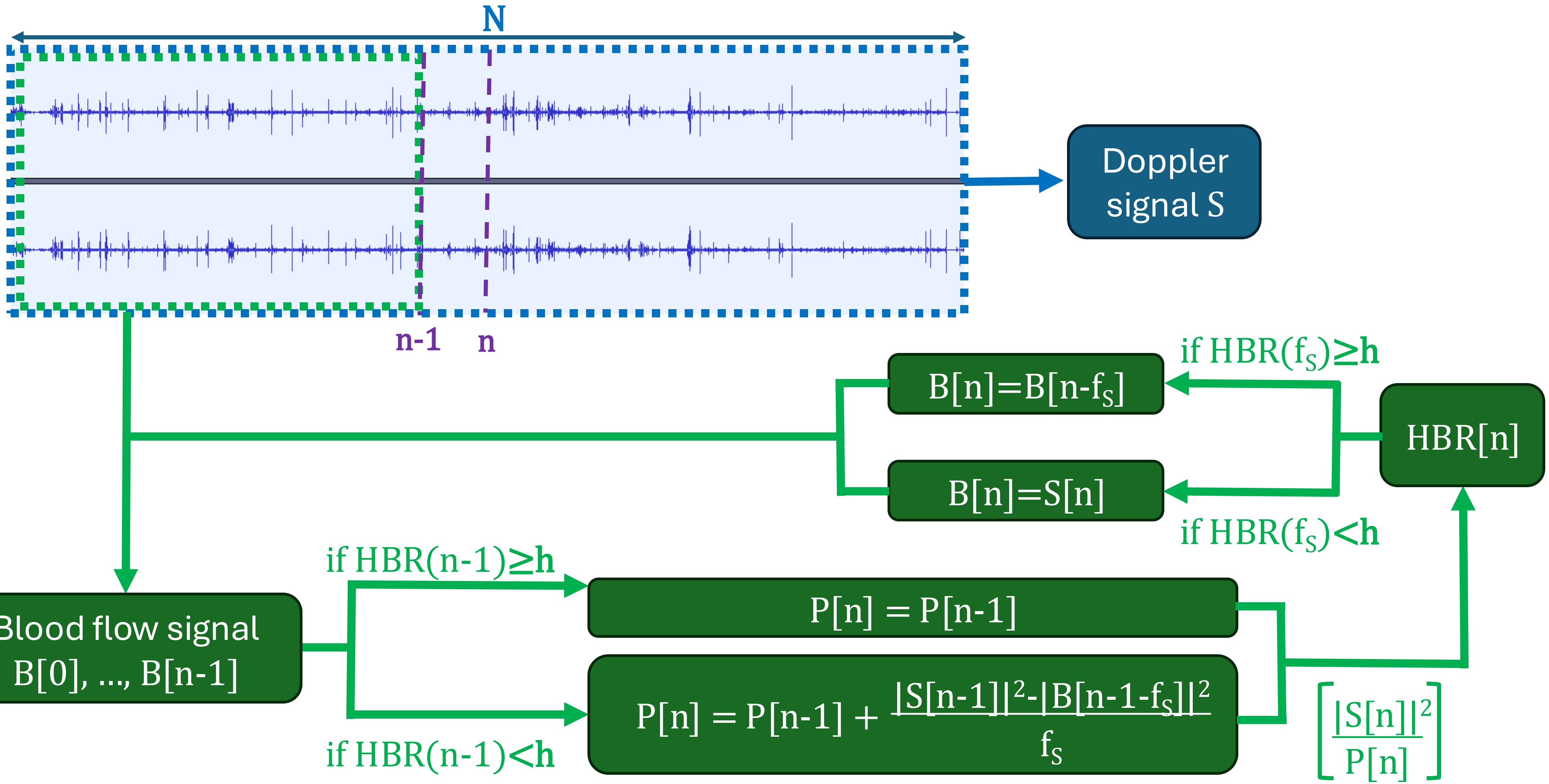
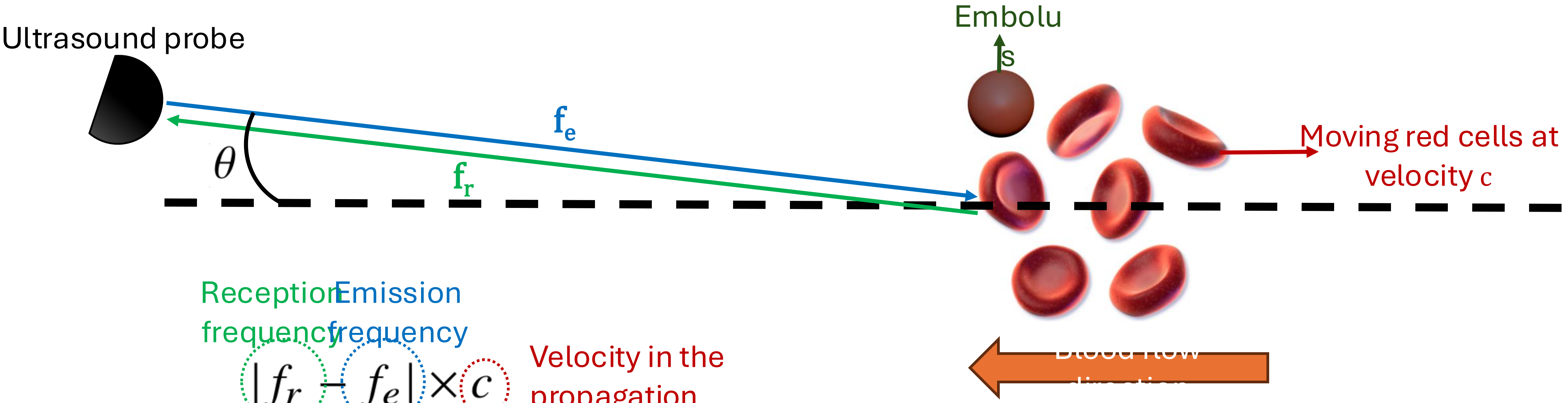


Figure – HBR computation **iteration** from Guépié et al. (2019)

Transcranial Doppler ultrasonography



Reception frequency

Emission frequency

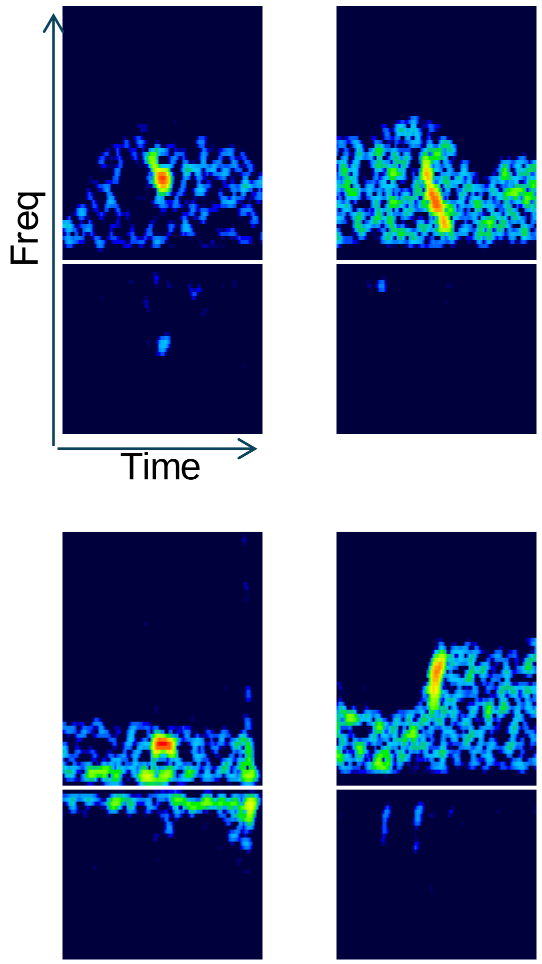
$$v_r = \frac{|f_r - f_e| \times c}{2 \times f_e \times cos(\theta)}$$

Velocity in the propagation medium

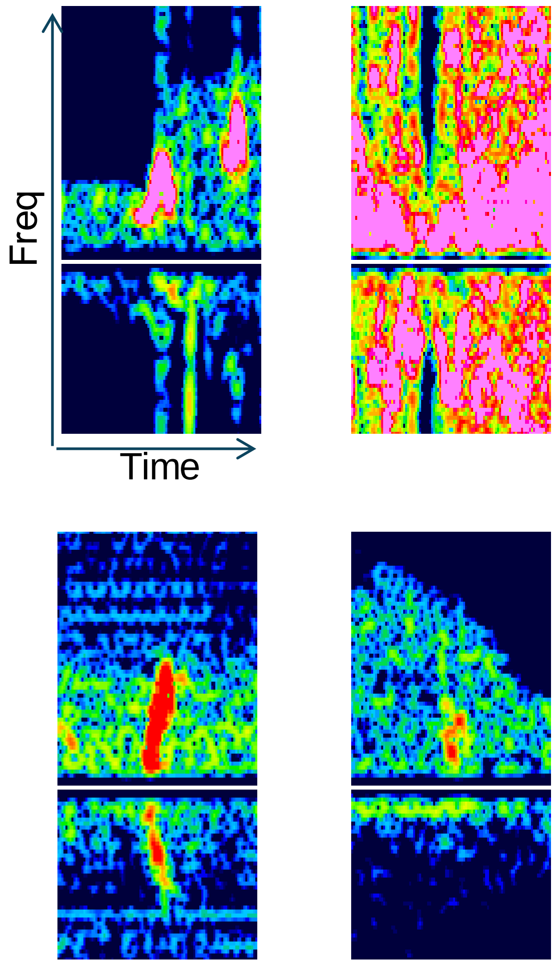
Doppler incidence angle

Challenges: data annotation

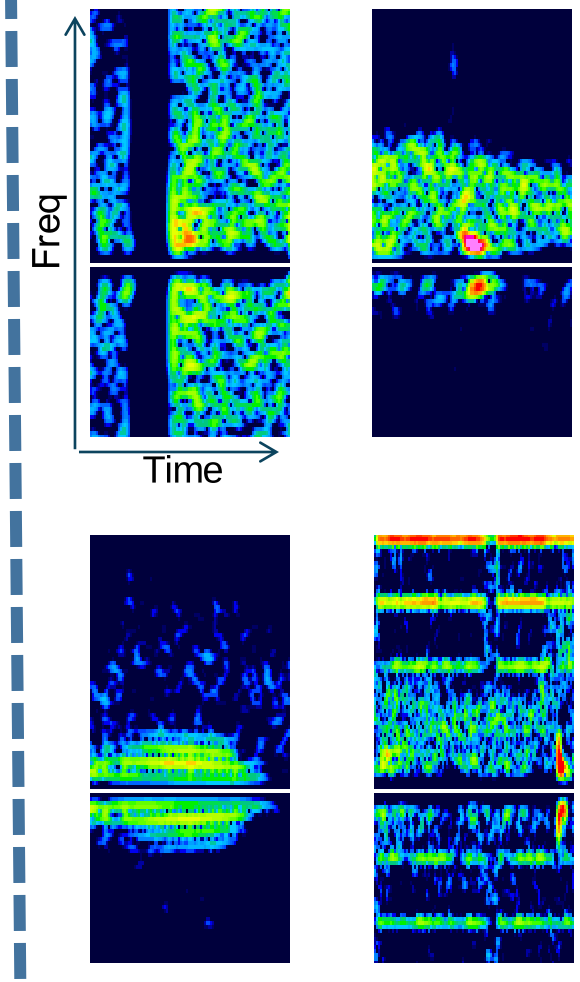
Solid Emboli



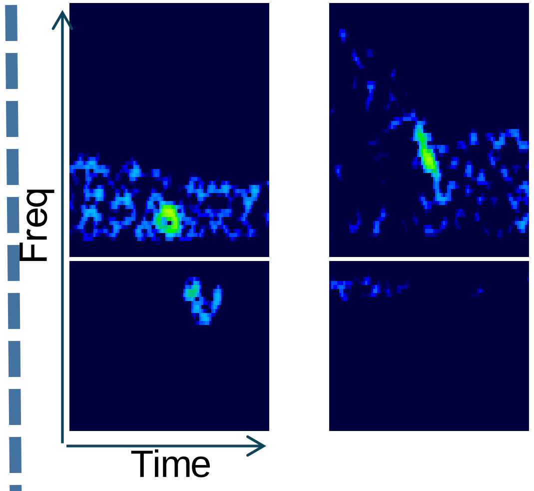
Gaseous Emboli



Artifact

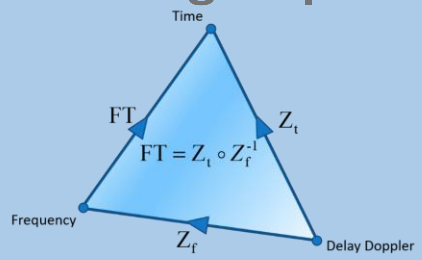


Difficult



Emboli classification

Signal processing



Georgiadis et al. (1994)

Aydin et al. (1999)

Aydin et al. (2004)

Marvasti et al. (2004)

Markus et al. (2005)

Chung et al. (2005)

Gencer et al. (2013)

Serbes et Aydin (2014)

Karahoca et al. (2015)

Imaduddin et al. (2020)

Machine learning

Darbellay et al. (2004)

Karahoca et al. (2007)

Chen et al. (2008)

Keunen et al. (2008)

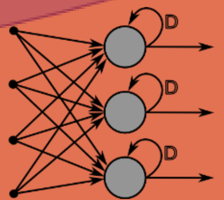
Sombune et al. (2016)

Guépié et al. (2017)

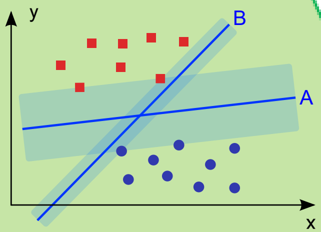
Guépié et al. (2019)

Tafsast et al. (2018)

Sombune et al. (2017)



Deep learning



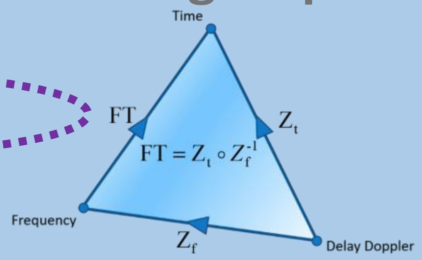
Emboli vs artifact

Solid emboli vs gaseous emboli

Solid emboli vs gaseous emboli vs artifact

Emboli classification

Signal processing



Georgiadis et al. (1994)

Aydin et al. (1999)

Aydin et al. (2004)

Marvasti et al. (2004)

Markus et al. (2005)

Chung et al. (2005)

Gencer et al. (2013)

Serbes et Aydin (2014)

Karahoca et al. (2015)

Imaduddin et al. (2020)

Machine learning

Darbellay et al. (2004)

Karahoca et al. (2007)

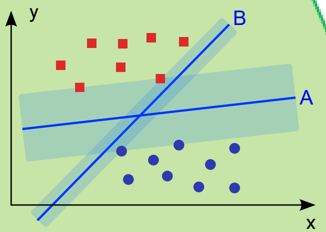
Chen et al. (2008)

Keunen et al. (2008)

Sombune et al. (2016)

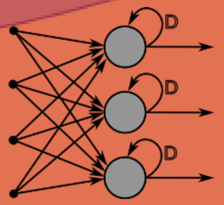
Guépié et al. (2017)

Guépié et al. (2019)



Tafsast et al. (2018)

Sombune et al. (2017)



Deep learning

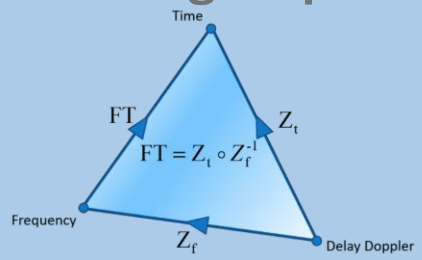
Emboli vs artifact

Solid emboli vs gaseous emboli

Solid emboli vs gaseous emboli vs artifact

Emboli classification

Signal processing



Georgiadis et al. (1994)

Aydin et al. (1999)

Aydin et al. (2004)

Marvasti et al. (2004)

Markus et al. (2005)

Chung et al. (2005)

Gencer et al. (2013)

Serbes et Aydin (2014)

Karahoca et al. (2015)

Imaduddin et al. (2020)

Machine learning

Darbellay et al. (2004)

Karahoca et al. (2007)

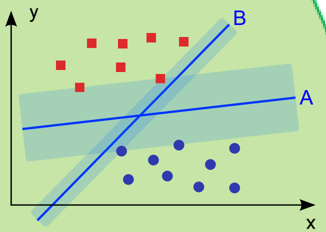
Chen et al. (2008)

Keunen et al. (2008)

Sombune et al. (2016)

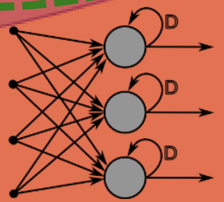
Guépié et al. (2017)

Guépié et al. (2019)



Tafsast et al. (2018)

Sombune et al. (2017)



Deep learning

Emboli vs artifact

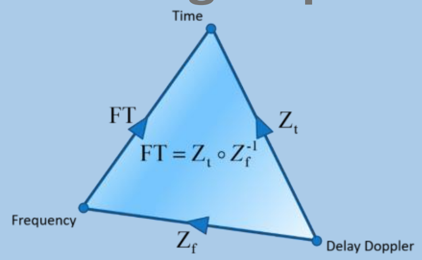
Solid emboli vs gaseous emboli

Solid emboli vs gaseous emboli vs artifact

Portable TCD data

Emboli classification

Signal processing



Georgiadis et al. (1994)

Aydin et al. (1999)

Aydin et al. (2004)

Marvasti et al. (2004)

Markus et al. (2005)

Chung et al. (2005)

Gencer et al. (2013)

Serbes et Aydin (2014)

Karahoca et al. (2015)

Imaduddin et al. (2020)

Machine learning

Darbellay et al. (2004)

Karahoca et al. (2007)

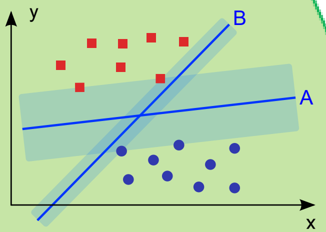
Chen et al. (2008)

Keunen et al. (2008)

Sombune et al. (2016)

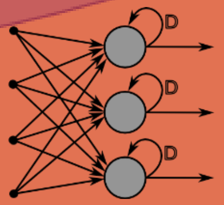
Guépié et al. (2017)

Guépié et al. (2019)



Tafsast et al. (2018)

Sombune et al. (2017)



Deep learning

First CNN for TCD emboli

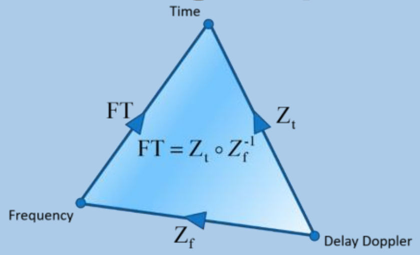
Emboli vs artifact

Solid emboli vs gaseous emboli

Solid emboli vs gaseous emboli vs artifact

Emboli classification

Signal processing



Georgiadis et al. (1994)

Aydin et al. (1999)

Aydin et al. (2004)

Marvasti et al. (2004)

Markus et al. (2005)

Chung et al. (2005)

Gencer et al. (2013)

Serbes et Aydin (2014)

Karahoca et al. (2015)

Imaduddin et al. (2020)

Machine learning

Darbellay et al. (2004)

Karahoca et al. (2007)

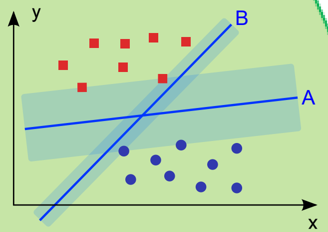
Chen et al. (2008)

Keunen et al. (2008)

Sombune et al. (2016)

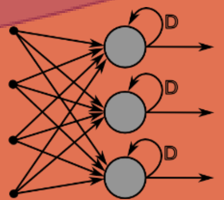
Guépié et al. (2017)

Guépié et al. (2019)



Tafsast et al. (2018)

Sombune et al. (2017)



Deep learning

Emboli vs artifact

Solid emboli vs gaseous emboli

Solid emboli vs gaseous emboli vs artifact

Emboli classification

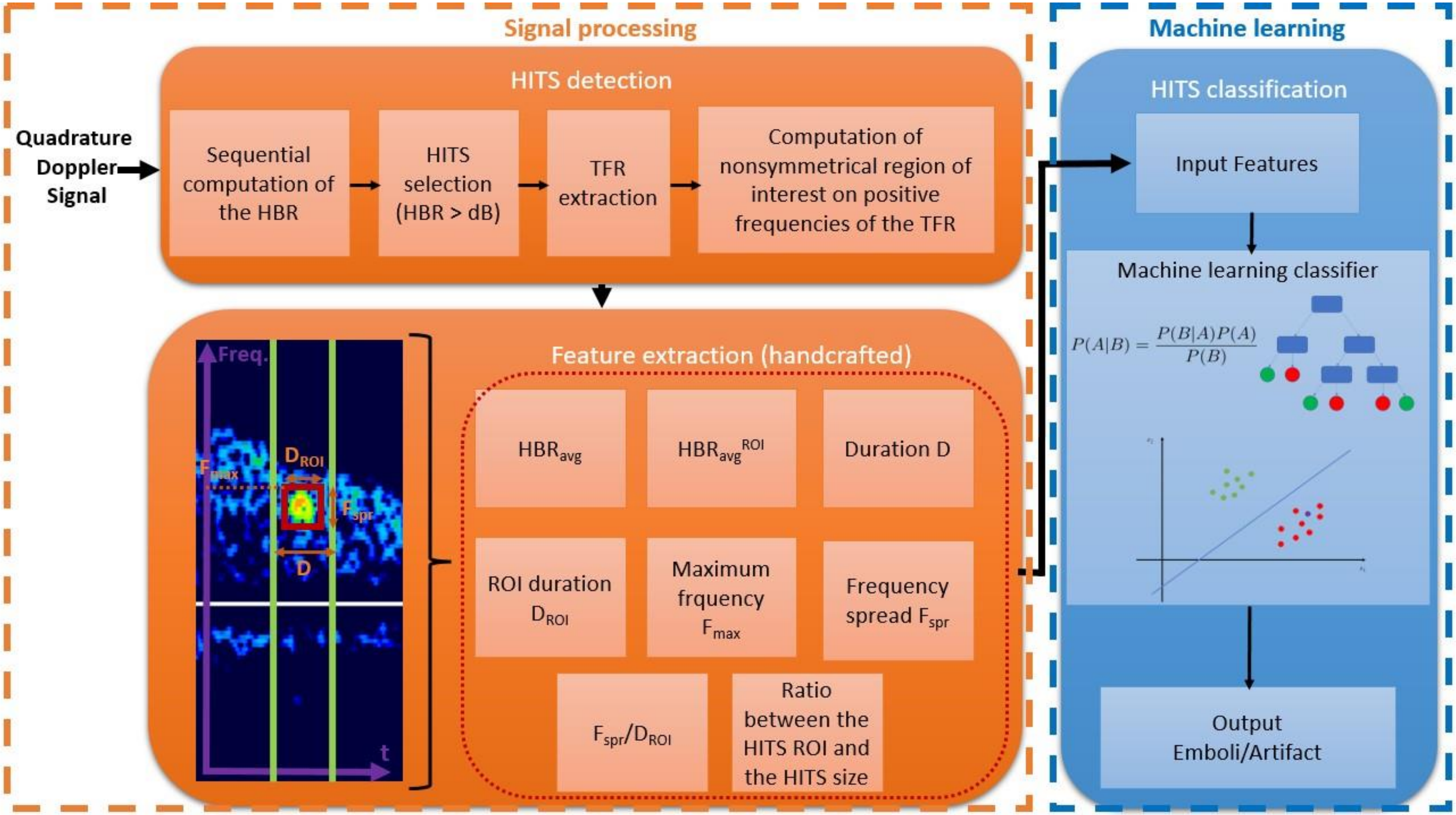


Figure – Proposed emboli detection and classification method of Guépié et al. (2019)

Emboli classification

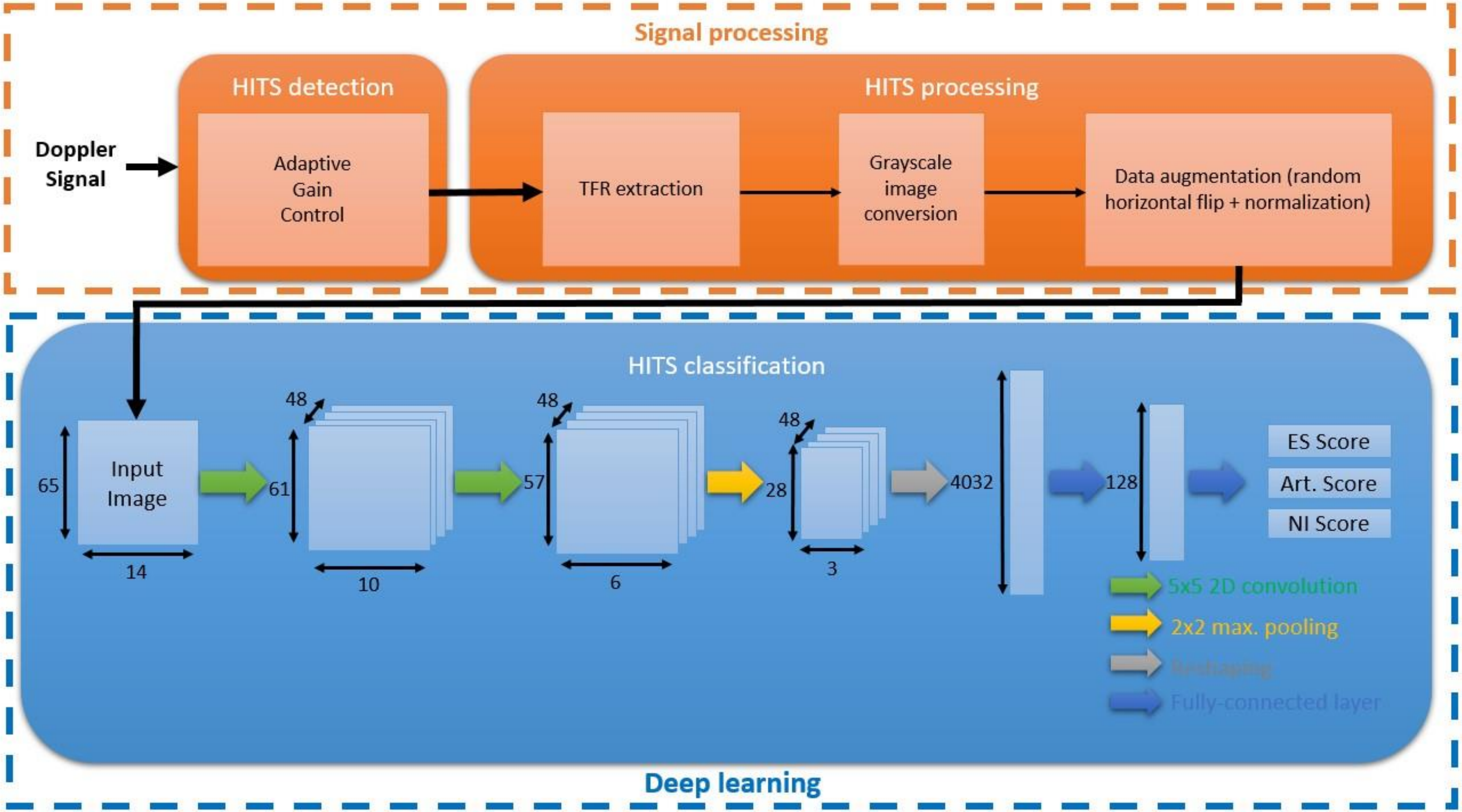
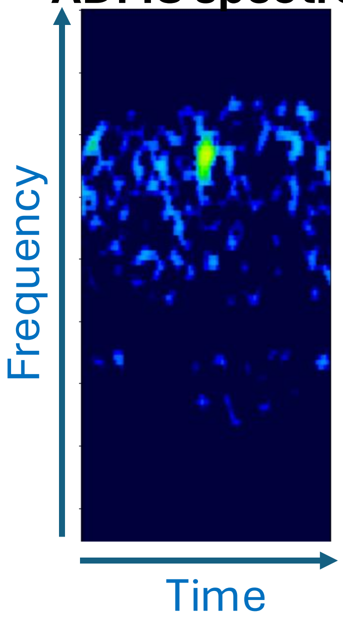


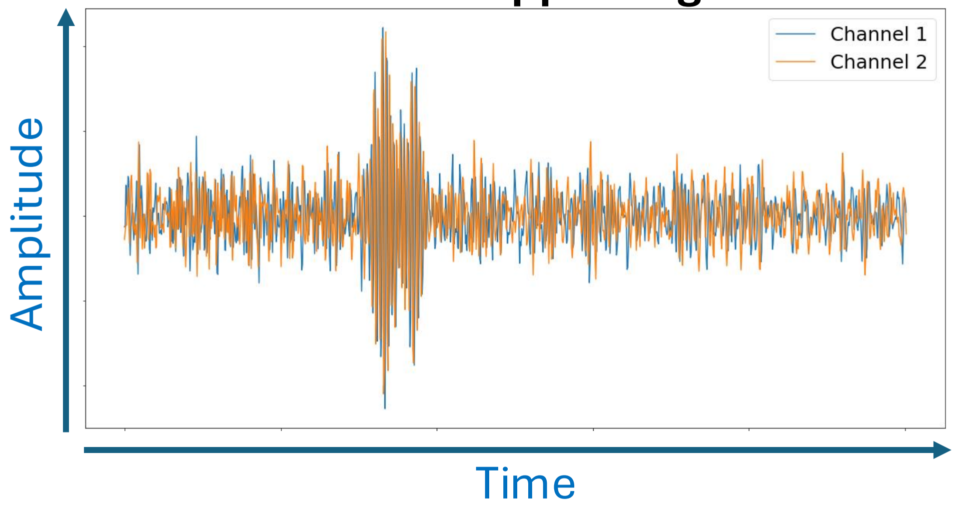
Figure – Proposed emboli HITS detection and classification by Sombune et al. (2017)

Other representations

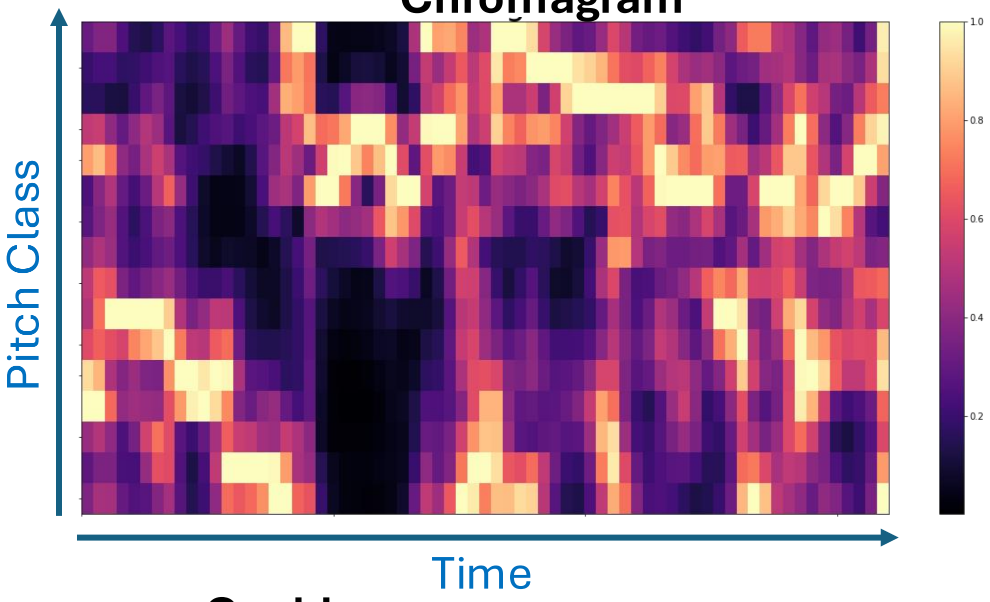
ADMS spectrogram



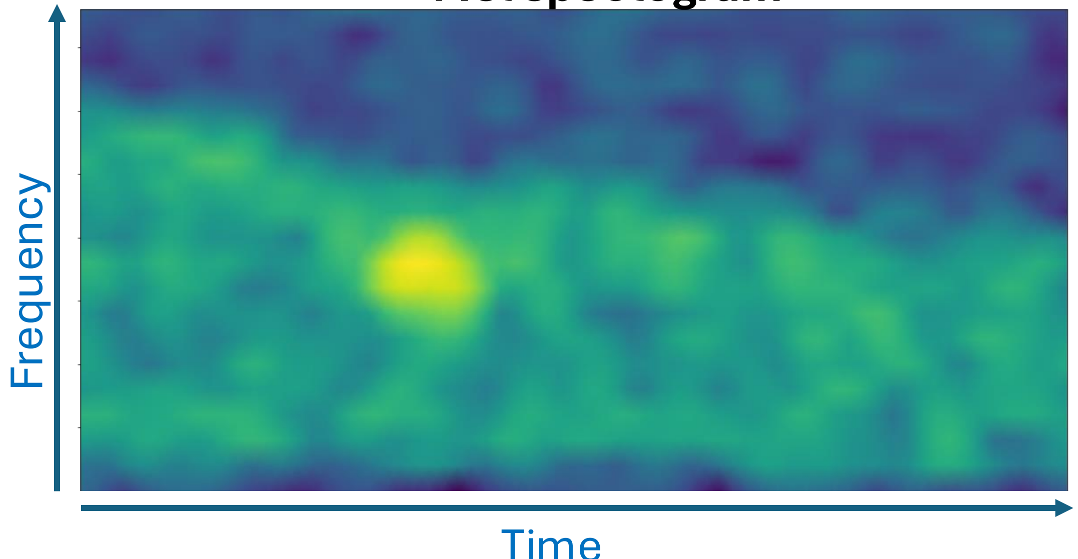
Raw Doppler Signal



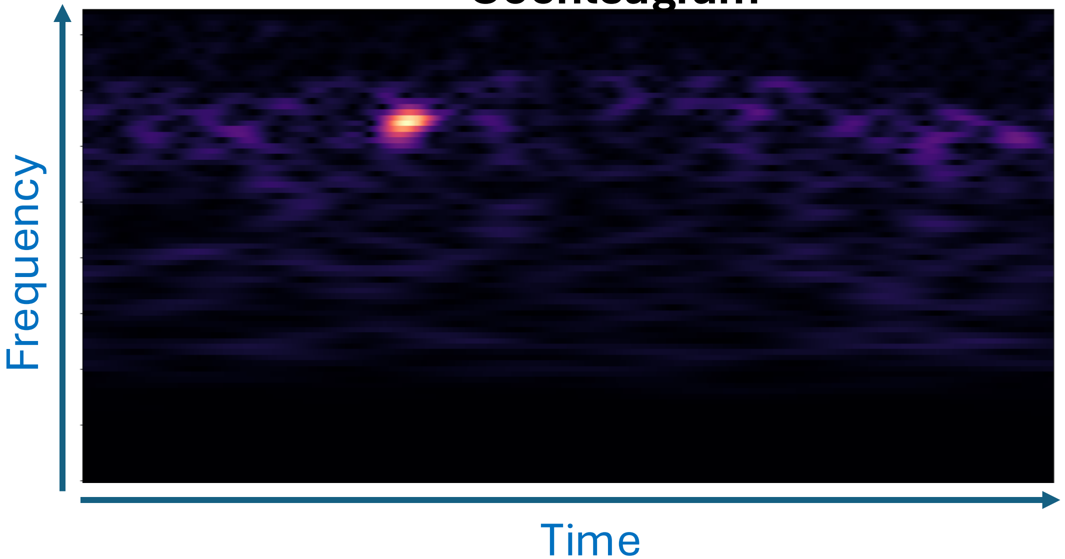
Chromagram



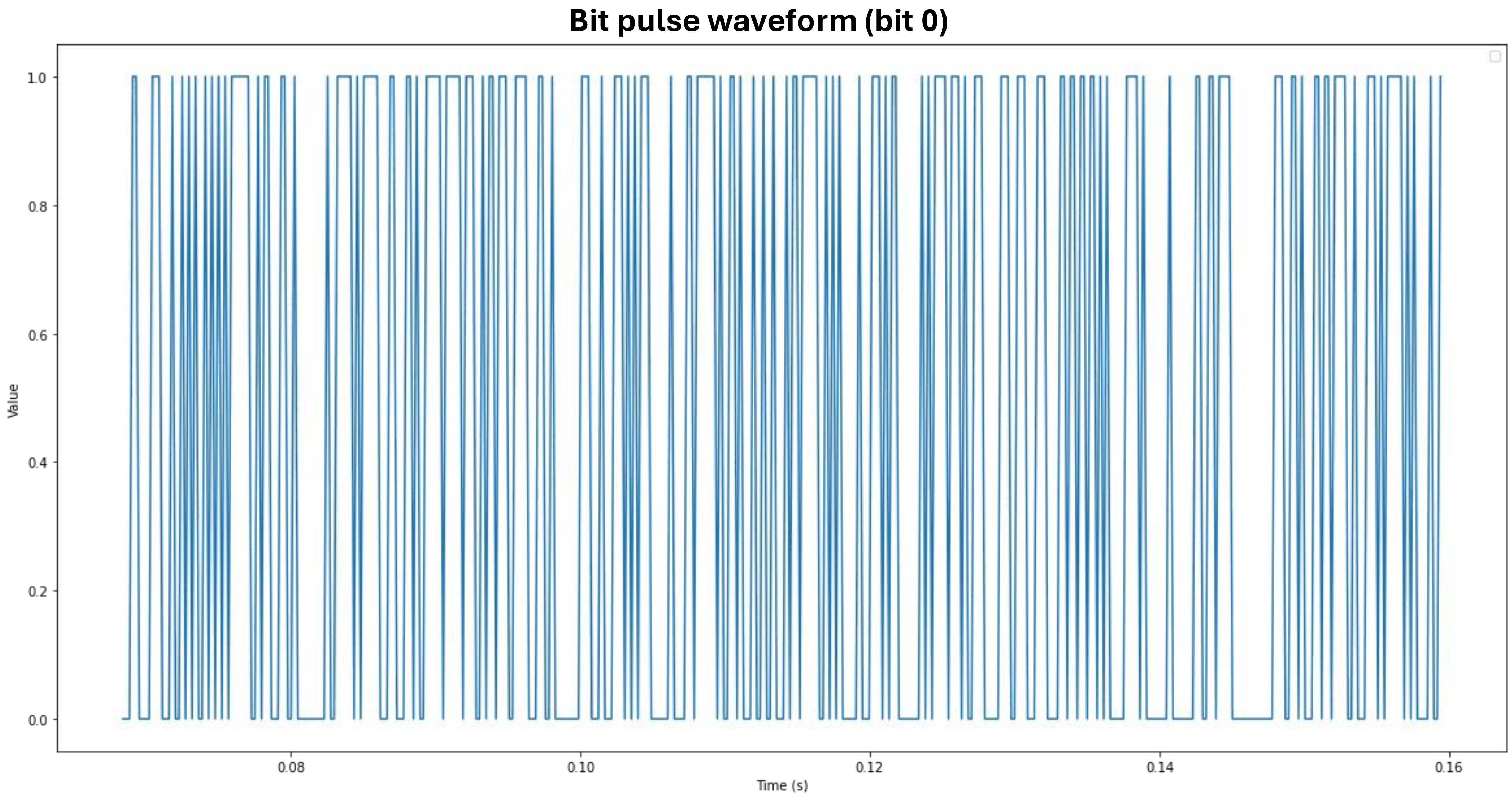
Mel spectogram



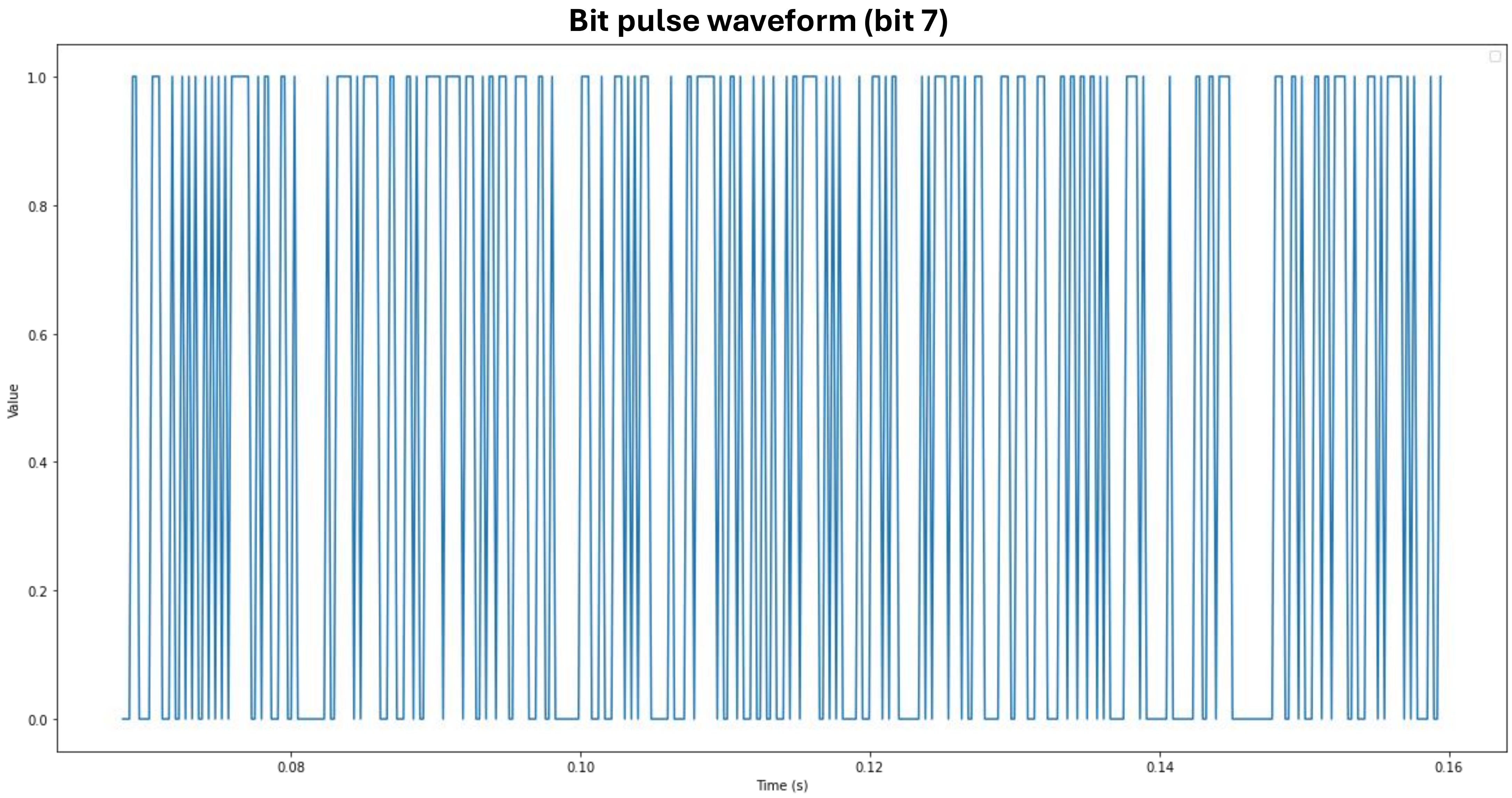
Cochleagram



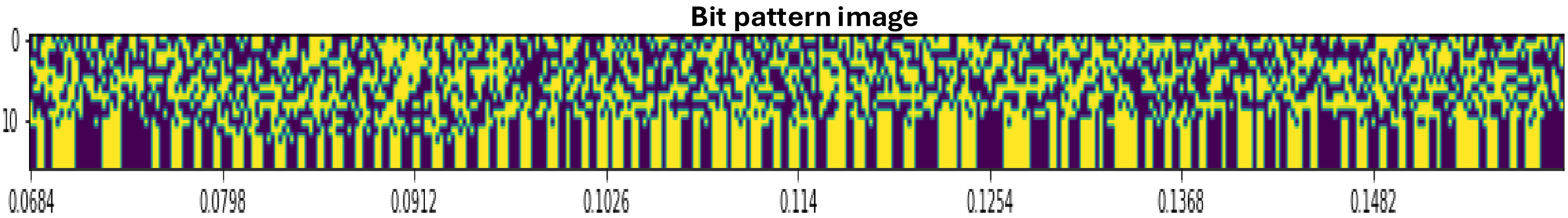
Other representations



Other representations



Other representations



Emboli classification

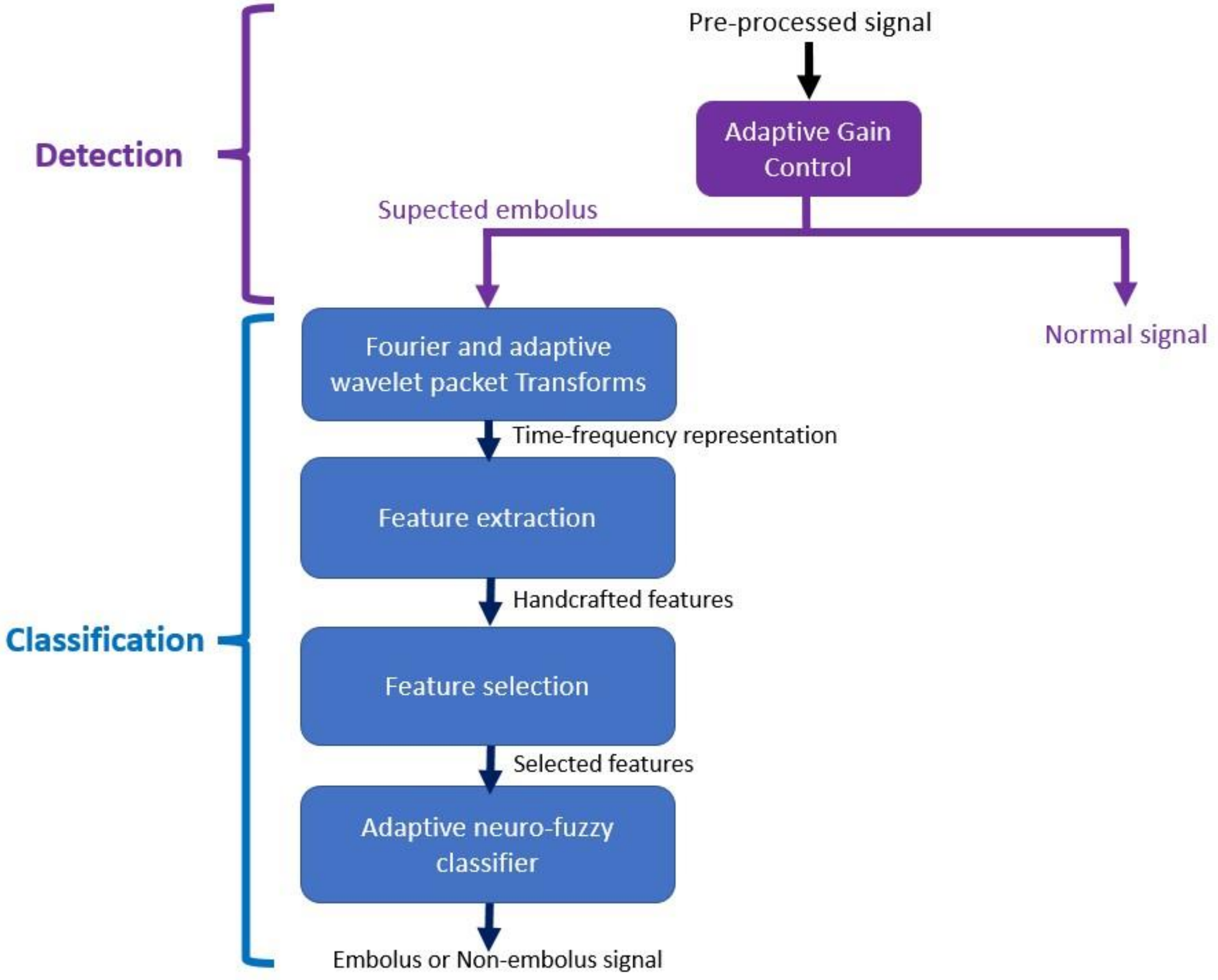


Figure – Proposed emboli ANFIS method by Sombune et al. (2016)

Emboli classification

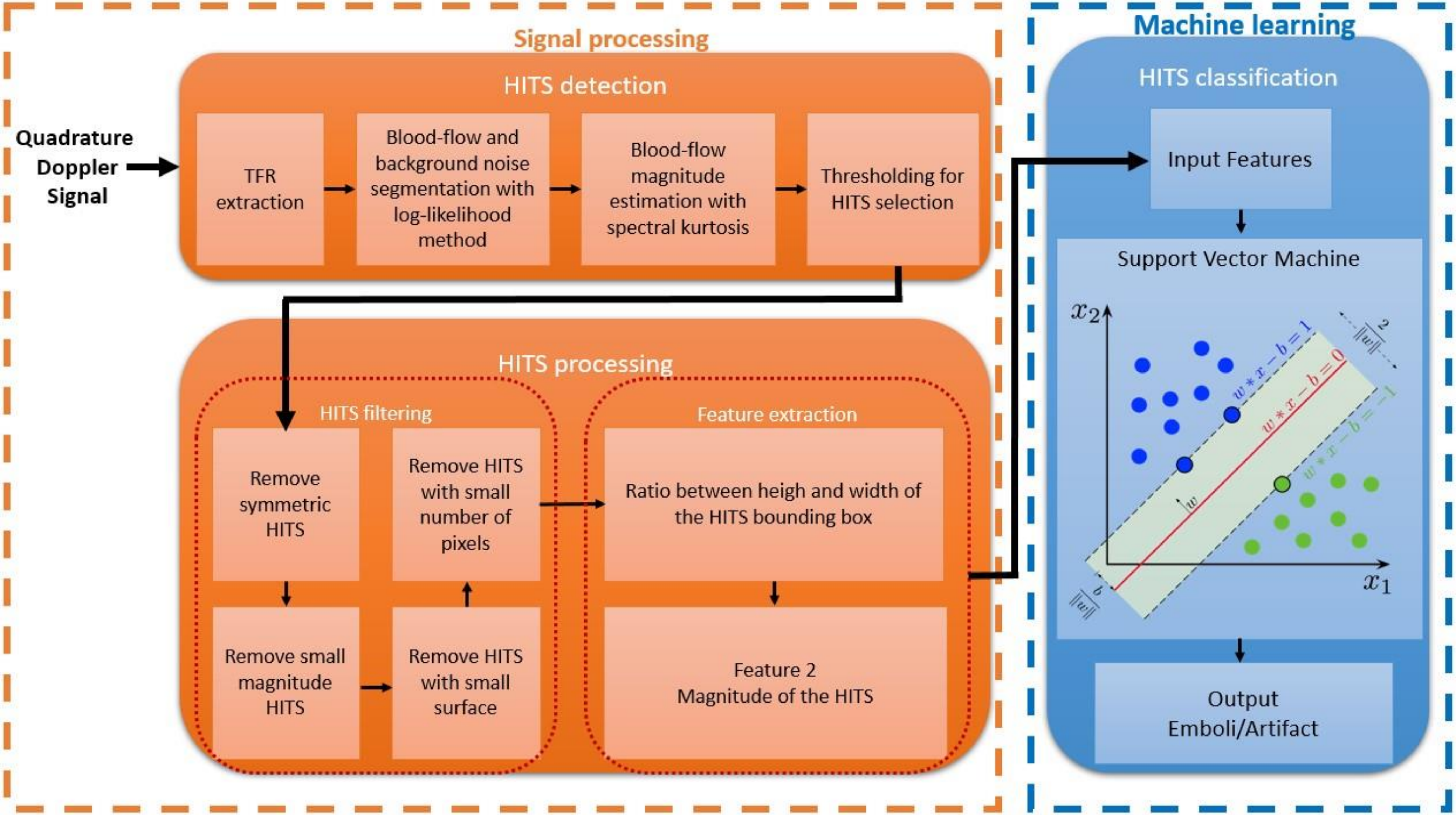


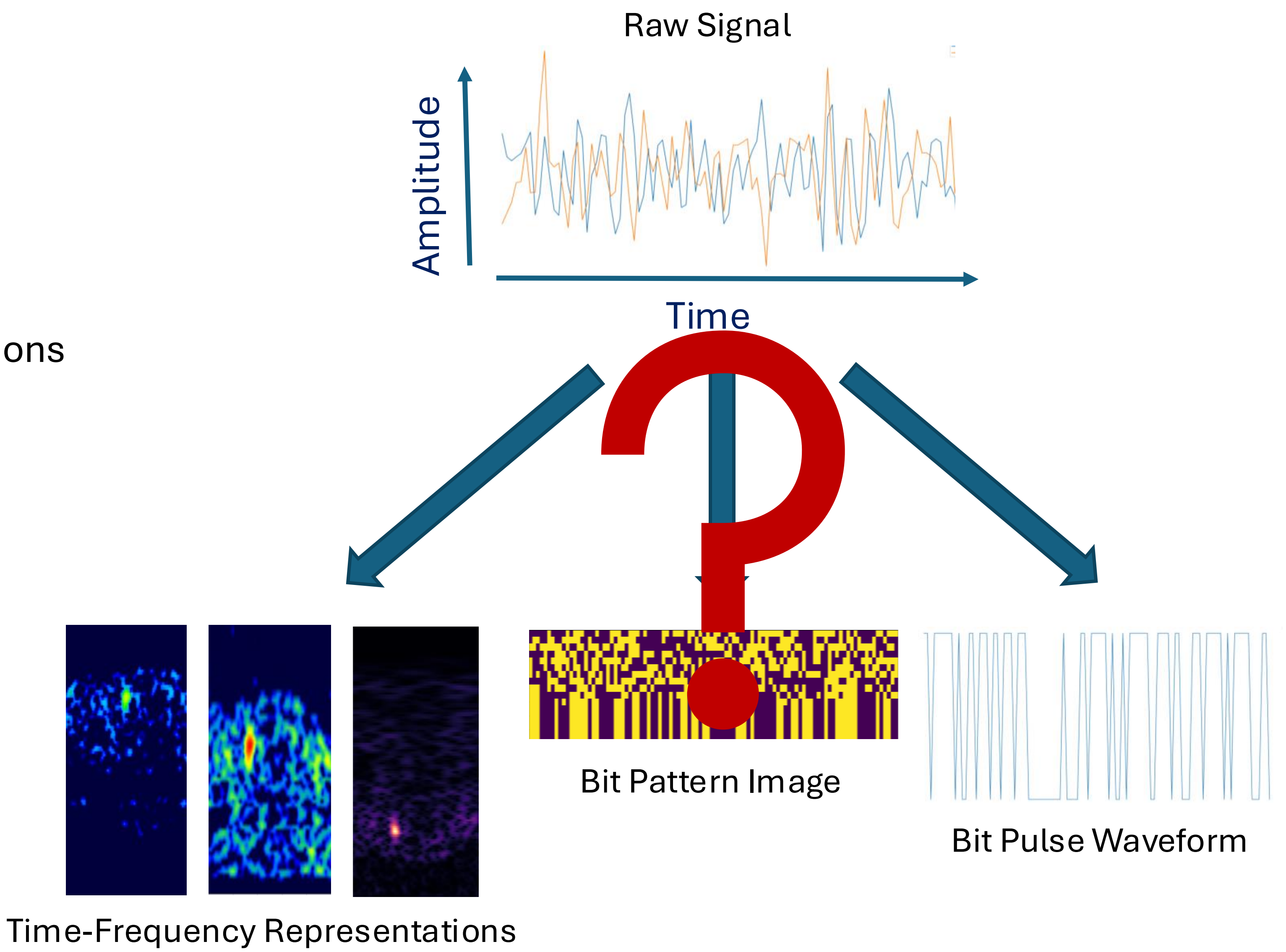
Figure – Proposed emboli detection and classification method of Guépié et al. (2017)

Emboli classification

Authors	Year	Fields	Methods	Classification	Advantages	Limitations
Markus et al.	2005	Signal Processing	-	SE vs GE	- Slightly better results than classical methods	- Use of a dual-frequency TCD - Non portable TCD
Sombune et al.	2017	Deep Learning	CNN	Artifact vs Embolus vs Normal	- No handcrafted features	- Non portable TCD - Not state-of-the-art performances
Guépié et al.	2017	Signal Processing Machine Learning	Likelihood segmentation Spectral Kurtosis, SVM	Artifact vs Embolus	- Good classification performance - Adapt to patients - Operator independent - Portable TCD	- Handcrafted features - No distinction between emboli
Tafsast et al.	2018	Deep Learning	CNN	SE vs GE	Good classification results	- In-vitro study
Guépié et al.	2019	Signal Processing Machine Learning	SVM, Naive Bayes, Decision Tree	Artifact vs Embolus	- State-of-the-art results - Adapt to patients - Operator independent - Sequential Method - Portable TCD	- Handcrafted features - No distinction between emboli

Challenges: optimal representation

- Temporal dependence.
- One modality, different representations
- Optimal representation ?
- Feature combination ?



Challenges: model compression

- Limited memory resources.
- Limited computation resources.
- Energy constraints.

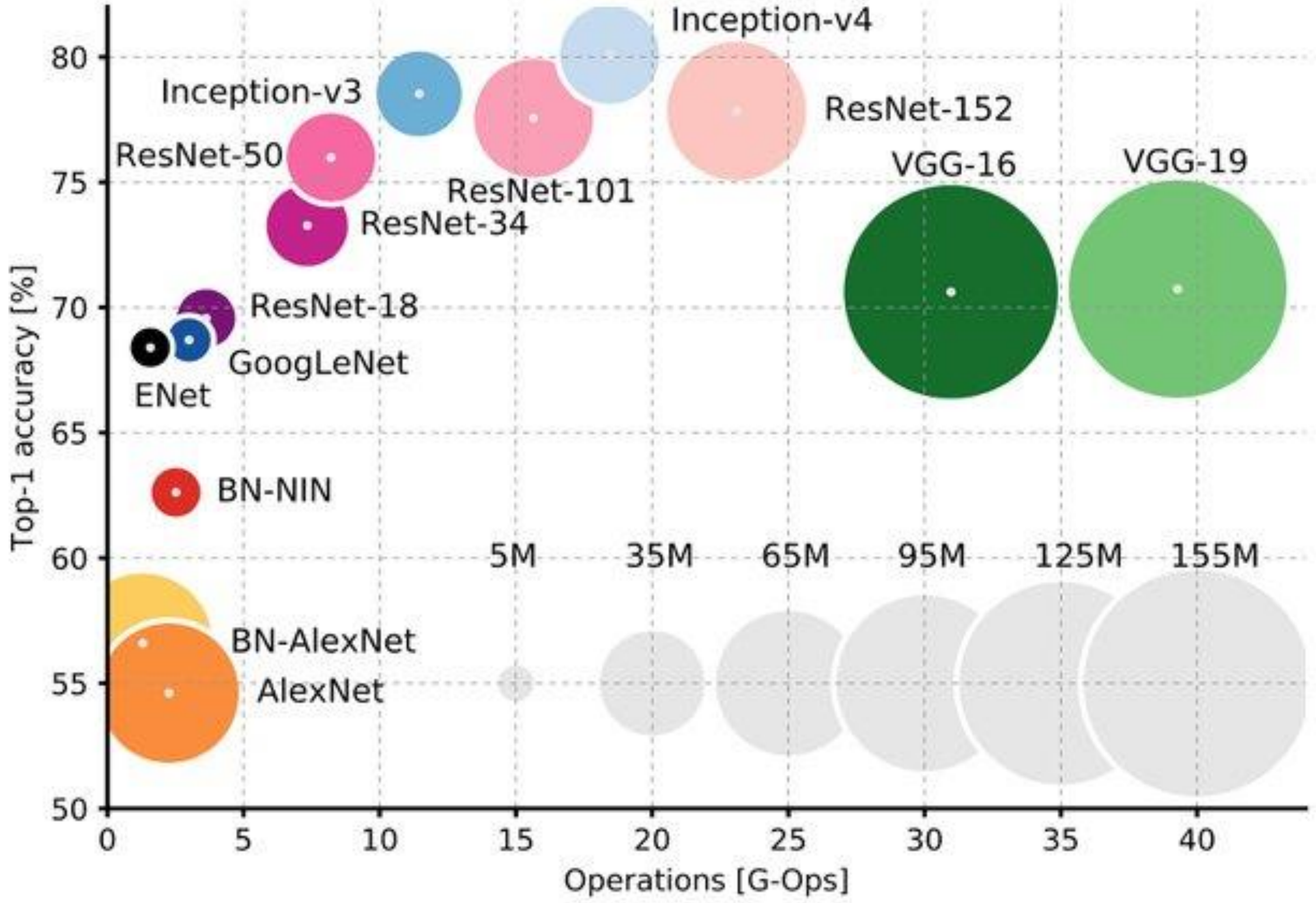


Figure – Classification accuracy based on the size and number of floating-point operations of different deep learning models (Abbas et al. 2021)

Challenges: limited resources

- Limited memory.
- Inference time.
- Energy consumption.

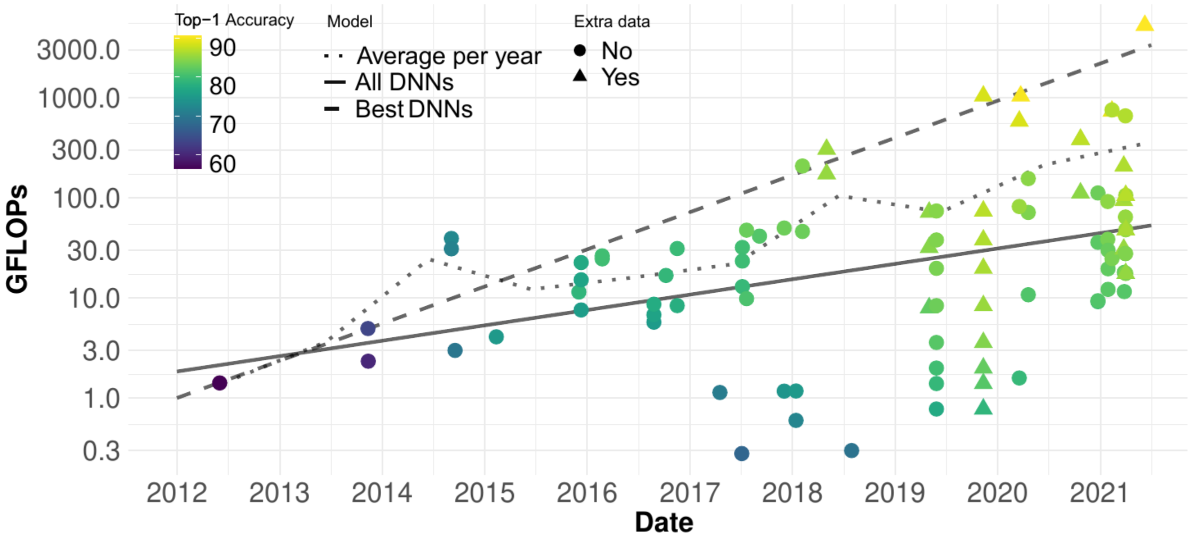


Figure – GFLOPs over the years. The dashed line is a linear fit (logarithmic y-axis) for the models with highest accuracy per year. Desislavov et Martinez-Plumed (2021).

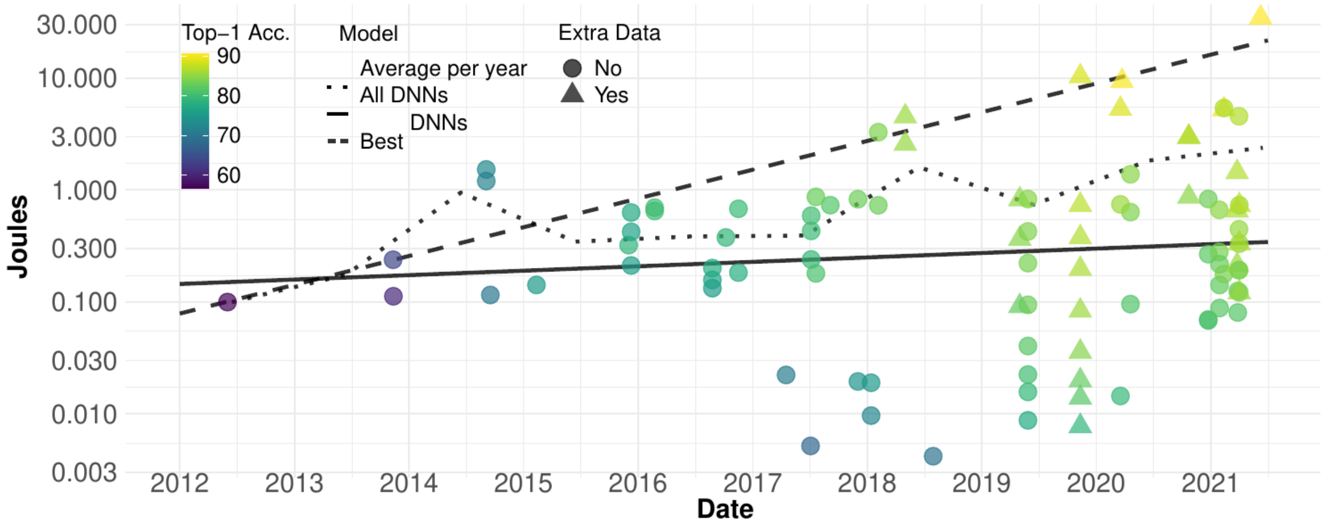


Figure – Estimated Joules of a forward pass (CV). The dashed line is a linear fit (logarithmic y-axis) for the models with highest accuracy per year. Desislavov et Martinez-Plumed (2021).

Contribution 1 : Semi-automatic data annotation

Possible solution to noisy-labels

Category		P1 Flexibility	P2 No Pre-train	P3 Full Exploration	P4 No Supervision	P5 Heavy Noise	P6 Complex Noise
Robust Loss Function		◯	◯	◯	◯	×	×
Robust Architecture	Noise Adaptation Layer	△	◯	◯	◯	×	×
	Dedicated Architecture	×	◯	◯	×	△	◯
Robust Regularization		◯	◯	◯	◯	△	△
Loss Adjustment	Loss Correction	◯	◯	◯	×	×	×
	Loss Reweighting	◯	◯	◯	◯	×	△
	Label Refurbishment	◯	◯	◯	△	×	△
Sample Selection		◯	◯	×	×	◯	△
Meta Learning	Fast Adaptation	◯	◯	◯	△	△	◯
	Learning to Update	◯	◯	◯	×	△	◯
Semi-supervised Learning		◯	△	◯	◯	◯	△

Figure – Table from Song et al. 2020. *O* means completely supported, △ means partially supported and × means not supported.

Input sample # classes

One-hot label

Model

k[th] element

$$\mathcal{L}_{GCE}(f(X), \overline{y}) = \sum_{k=1}^{K} \frac{\overline{y}_k - f_k(X)^q}{q}$$

$$\xrightarrow{q \to 0} \mathcal{L}_{CE}(f(X), \overline{y})$$

Noise sensitive

$$\xrightarrow{q \to 1} \mathcal{L}_{MAE}(f(X), \overline{y})$$

Noise tolerant

Noisy-labels

Category		Method	P1	P2	P3	P4	P5	P6
Robust Loss Function		*Robust MAE*	O	O	O	O	×	×
		Generalized Cross Entropy	O	O	O	O	×	×
		Symmetric Cross Entropy	O	O	O	O	×	×
		Curriculum Learning	O	O	O	×	O	△
Robust Architecture	Noisy Adaptation Layer	*Webly Learning*	△	×	O	O	×	×
		Noise Model	△	O	O	O	×	×
		Dropout Noise Model	△	O	O	O	×	×
		S-model	△	O	O	O	×	×
		C-model	△	O	O	O	×	O
		NLNN	△	O	O	O	×	×
	Dedicated Architecture	*Probablistic Noise Model*	×	×	O	×	△	O
		Masking	×	O	O	×	△	O
		Contrastive-Additive Noise Network	×	O	O	O	△	O
Robust Regularization		*Adversarial Training*	O	O	O	O	△	△
		Label Smoothing	O	O	O	O	△	△
		Mixup	O	O	O	O	△	△
		Bilevel Learning	O	O	O	×	△	△
		Annotator Confusion	O	×	O	O	△	△
		Pre-training	O	×	O	O	△	△
Loss Adjustment	Loss Correction	*Backward Correction*	O	O	O	×	×	×
		Forward Correction	O	O	O	×	×	×
		Gold Loss Correction	O	×	O	×	×	×
	Loss Reweigting	*Importance Reweighting*	O	O	O	O	×	△
		Active Bias	O	O	O	O	×	△
	Label Refurbishment	*Bootstrapping*	O	O	O	×	×	△
		Dynamic Bootstrapping	O	O	O	O	×	△
		D2L	O	O	O	O	×	△
		SELFIE	O	O	O	×	O	△

Category		Method	P1	P2	P3	P4	P5	P6
Sample Selection		*Decouple*	O	O	×	O	×	△
		MentorNet	×	×	×	×	O	△
		Co-teaching	O	O	×	×	O	△
		Co-teaching+	O	O	×	×	O	△
		Iterative Detection	O	O	×	O	O	△
		ITLM	O	O	×	×	O	△
		INCV	O	O	×	O	O	△
Meta Learning	Fast Adaption	*Meta-Regressor*	O	O	O	×	O	O
		MLNT	O	O	O	O	×	O
	Learning to Update	*Knowledge Distillation*	O	×	O	×	△	O
		L2LWS	×	O	O	×	△	O
		CWS	×	O	O	×	△	O
		Automatic Reweighting	O	O	O	×	△	O
		Meta-Weight-Net	△	O	O	×	△	O
		Data Coefficients	O	O	O	×	O	O
Semi-supervised Learning		*Label Aggregation*	O	×	O	×	×	△
		Two-Stage Framework	O	×	O	O	O	△
		SELF	O	O	O	O	O	△
		DivideMix	O	O	O	O	O	△

Figure – Table from Song et al. 2020. *O* means completely supported, △ means partially supported and × means not supported.

OPF-semi

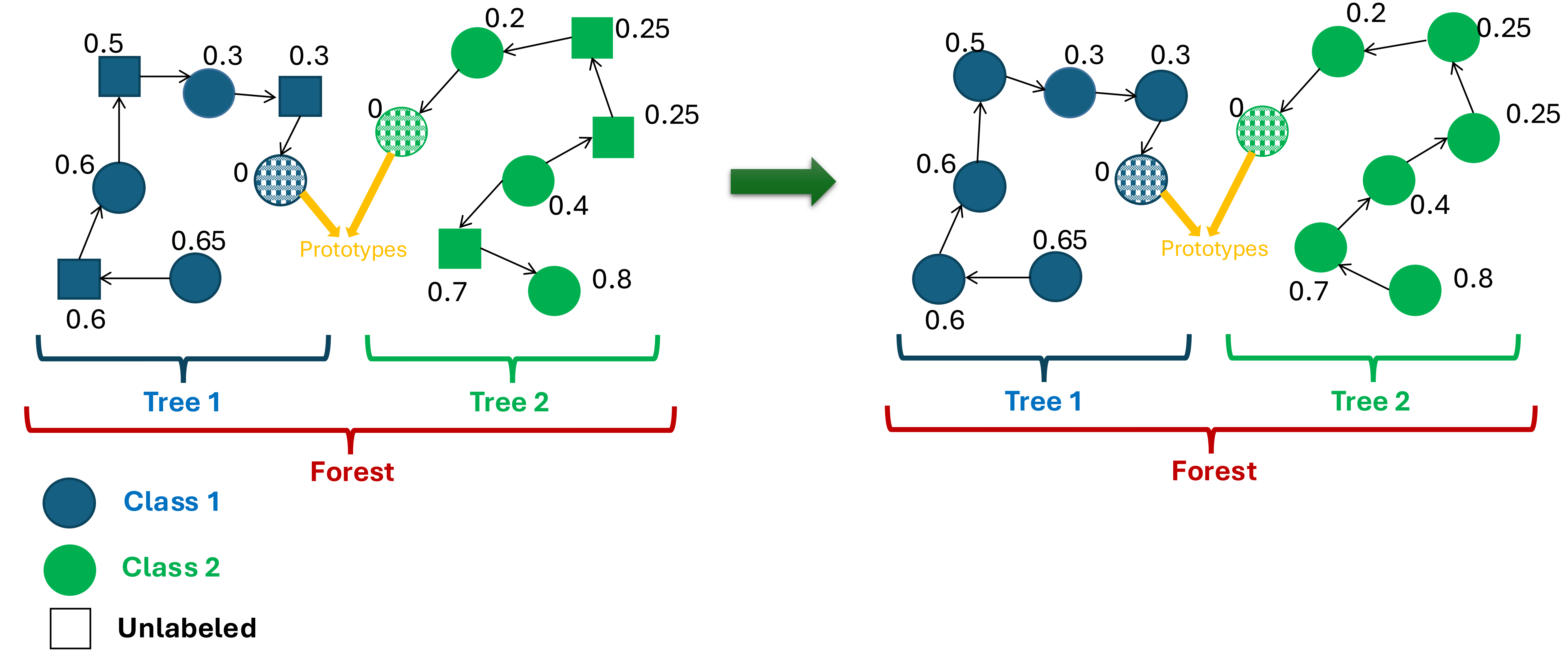


Figure – Semi-supervised optimum path forest (OPF-semi) (**Amorim et al., 2014**)

Co-ranking framework: local quality

$$Q_L^i(k_s,k_t) = \frac{1}{2\times k_s \times N}\times\sum_{j=1}^{N}\left(\mu_t(R_{ij},r_{ij},k_t)\times\mu_s(R_{ij},r_{ij},k_s)+\mu_t(R_{ji},r_{ji},k_t)\times\mu_s(R_{ji},r_{ji},k_s)\right)$$

Size of the neighborhood to consider

Rank significance

$$\mu_s(R_{ij},r_{ij},k_s) = \begin{cases} 1 \text{ if } R_{ij} \leq k_s \text{ or } r_{ij} \leq k_s \\ 0 \text{ else} \end{cases}$$

Rank error tolerance

$$\mu_t(R_{ij},r_{ij},k_t) = \begin{cases} 1 \text{ if } |R_{ij} - r_{ij}| \leq k_t \\ 0 \text{ else} \end{cases}$$

Size of the tolerated ranks errors

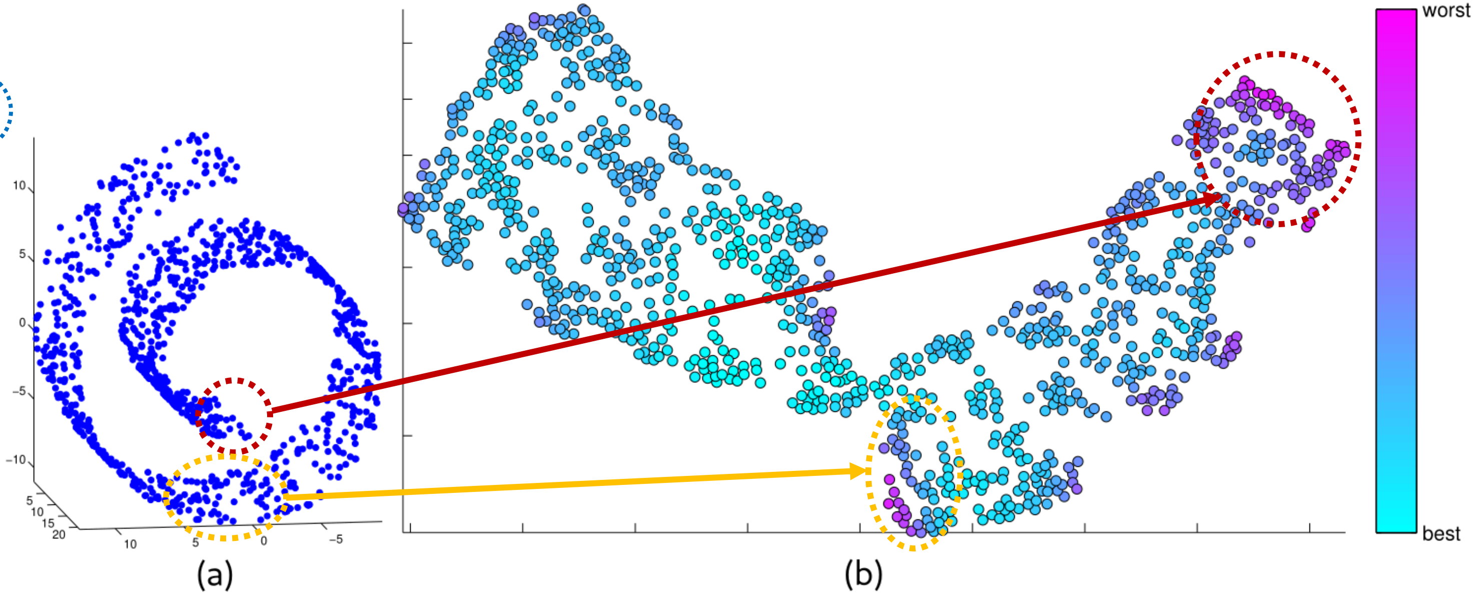


Figure – Illustration of the local quality metric on the Swiss roll benchmark data (**Lueks et al., 2011**).

Global pipeline

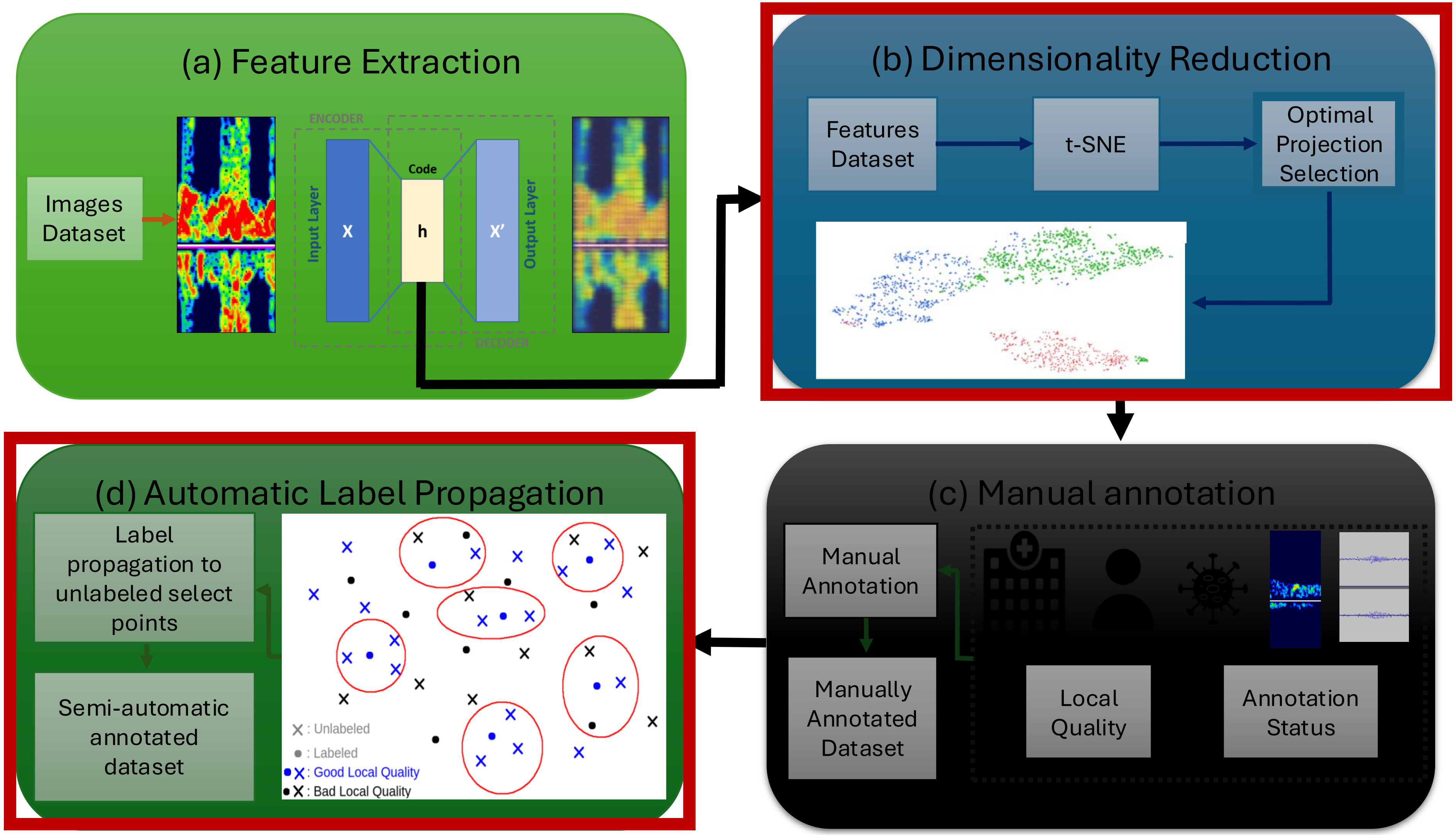


Figure – Global pipeline of our proposed semi-automatic data annotation approach.

Propagación semiautomática de etiquetas: Hipótesis

- Basado en un enfoque **K-nearest neighbors** (**KNN**).
- **Tres hipótesis**:
  - Hipótesis de estructura/cluster[1].
  - Conservación de las estructuras locales.
  - Cobertura del espacio de anotación.

Propagación de etiquetas Local Quality (**LQ**) K-Nearest Neighbor (**KNN**): **LQ-KNN**

[1] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, eds.Semi-supervised learning. Adaptive computation and machine learning. OCLC: ocm64898359. Cambridge, Mass: MIT Press, 2006. 508 pp.isbn: 978-0-262-03358-9

Diferentes etapas

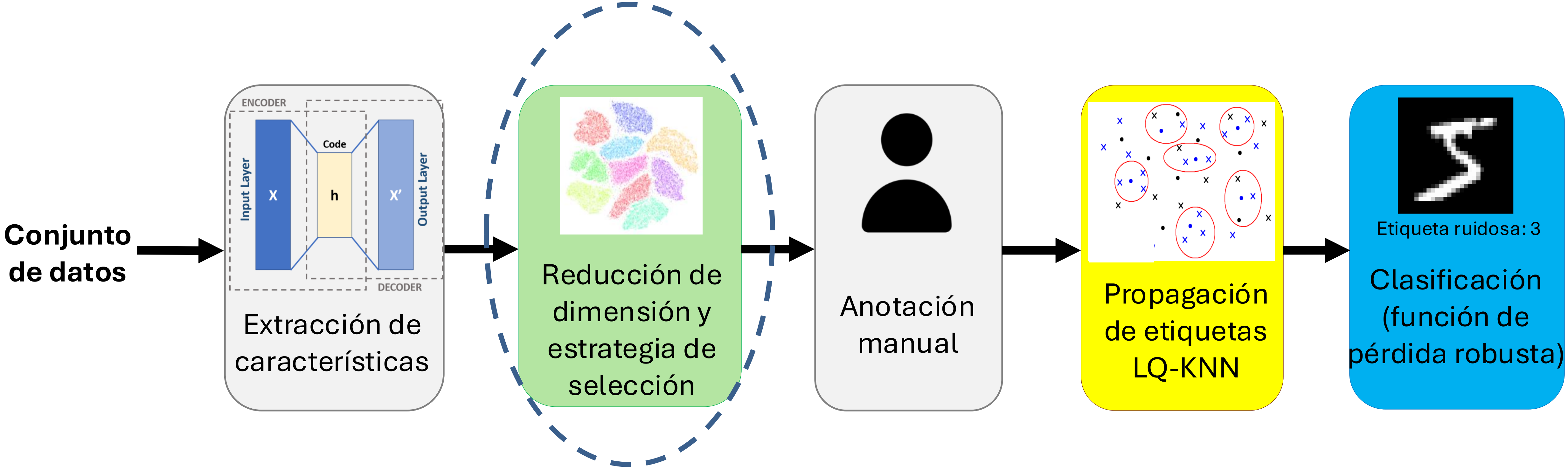
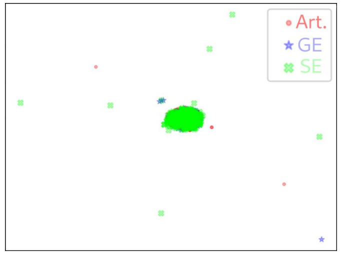
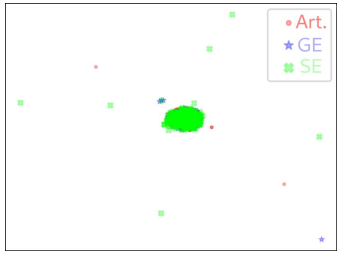


Figura - Método semiautomático de anotación de datos

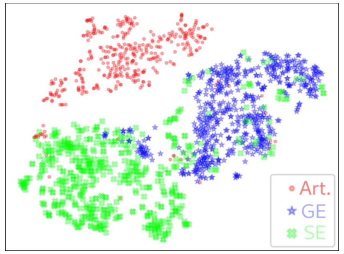
Reducción de dimensionalidad



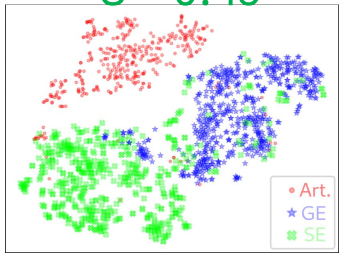
S = -0.56



Cálculo del
Silhouette Score S



S = 0.49

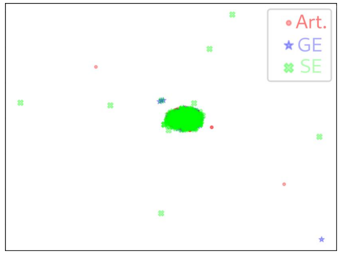


Diferentes
proyecciones 2D
calculadas

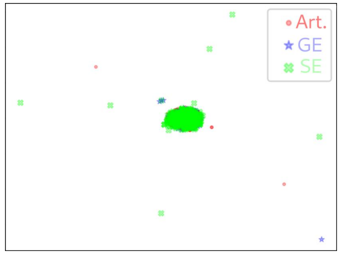
Silhouette score
S entre -1 y 1

Conjunto de puntos separados y
agrupados de manera compacta

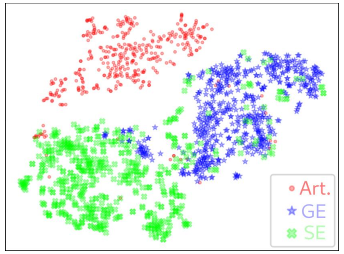
Reducción de dimensionalidad



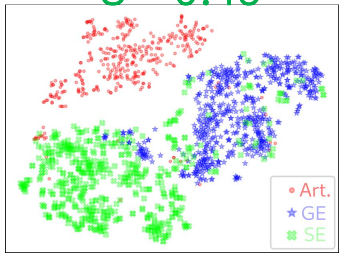
S = -0.56



Cálculo del
Silhouette Score *S*



S = 0.49

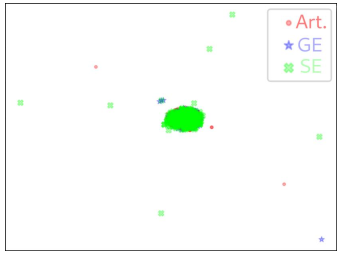


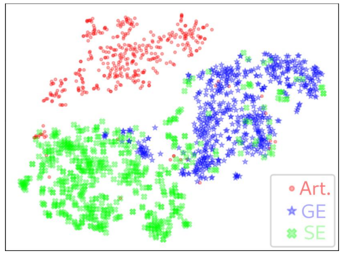
Diferentes
proyecciones 2D
calculadas

Silhouette score
S entre -1 y 1

Conjunto de puntos
se mezclan entre sí.

Reducción de dimensionalidad

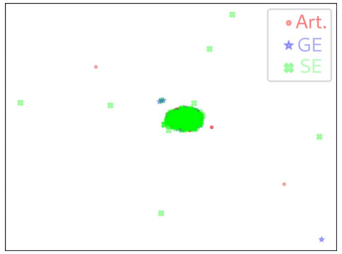




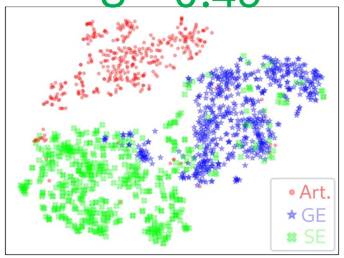
Diferentes proyecciones 2D calculadas

Cálculo del Silhouette Score *S*

S = -0.56



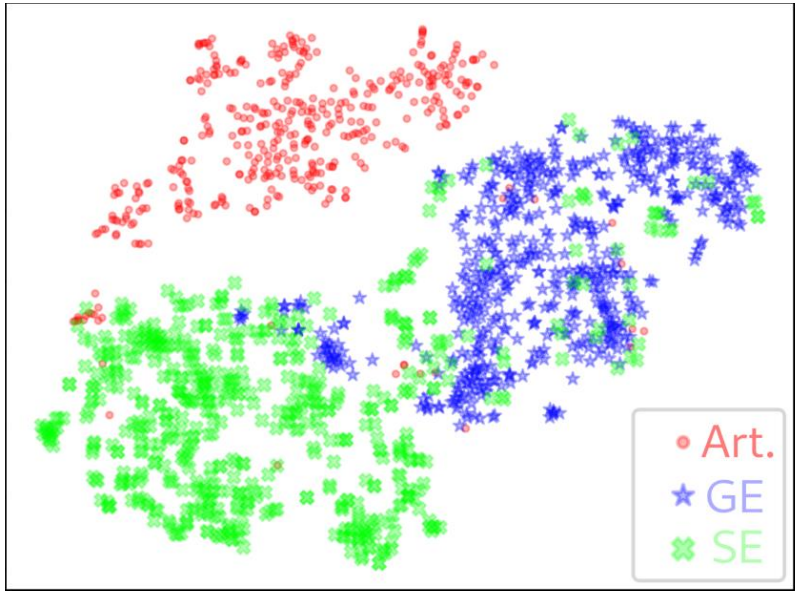
S = 0.49



Silhouette score *S* entre -1 y 1

Mejor Silhouette Score

S = 0.49



Proyección seleccionada para la anotación manual y la propagación de etiquetas

Proposed pipeline

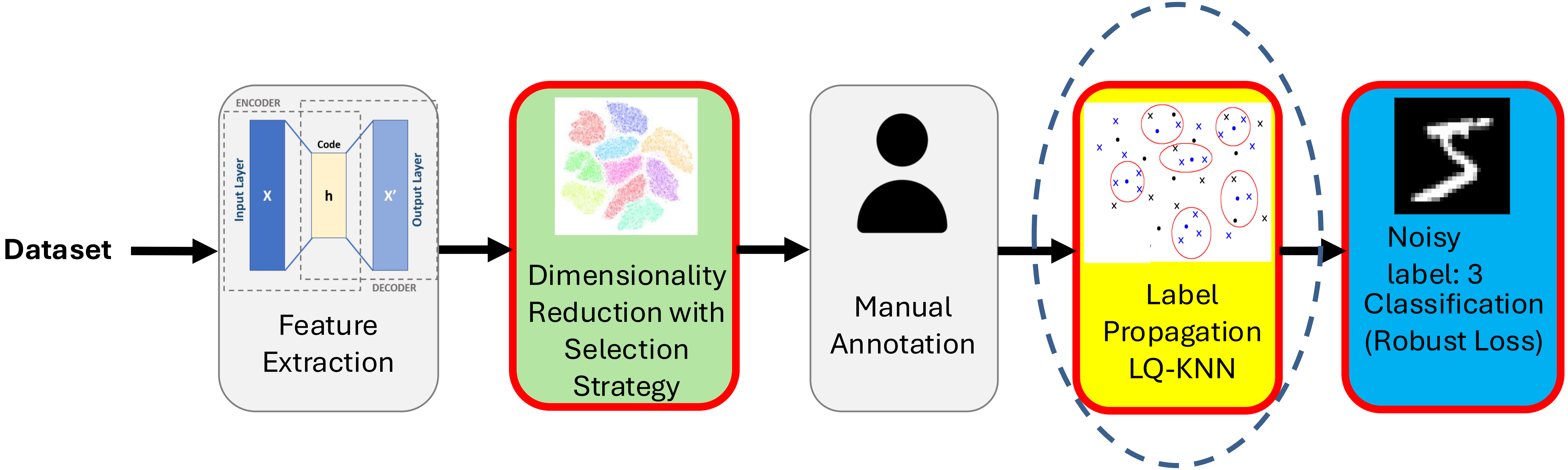


FIGURE - Semi-Automatic Data Annotation Method

Contribution 1.a.: Semi-automatic data annotation method **state-of-the-art comparison**

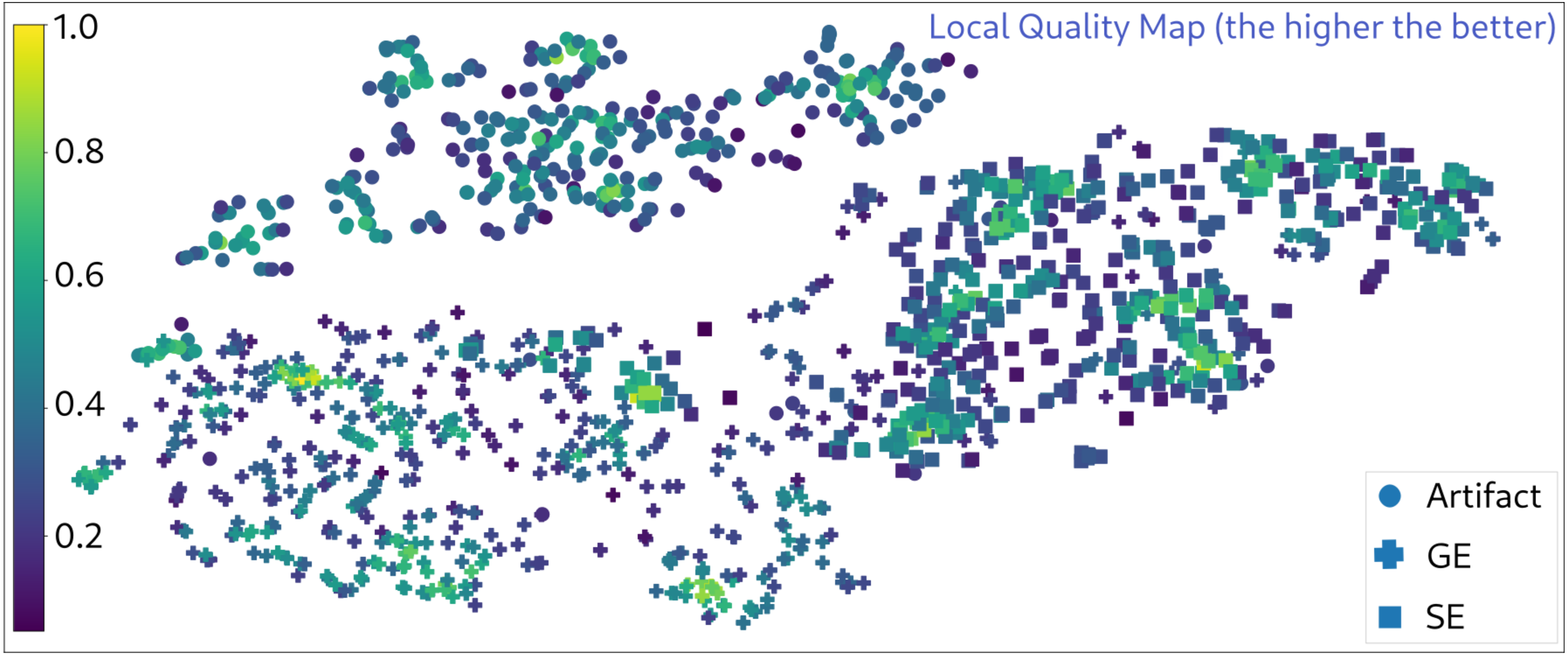


Figure - Local Quality Map of the unlabeled samples

Contribution 1.a.: Semi-automatic data annotation method **state-of-the-art comparison**

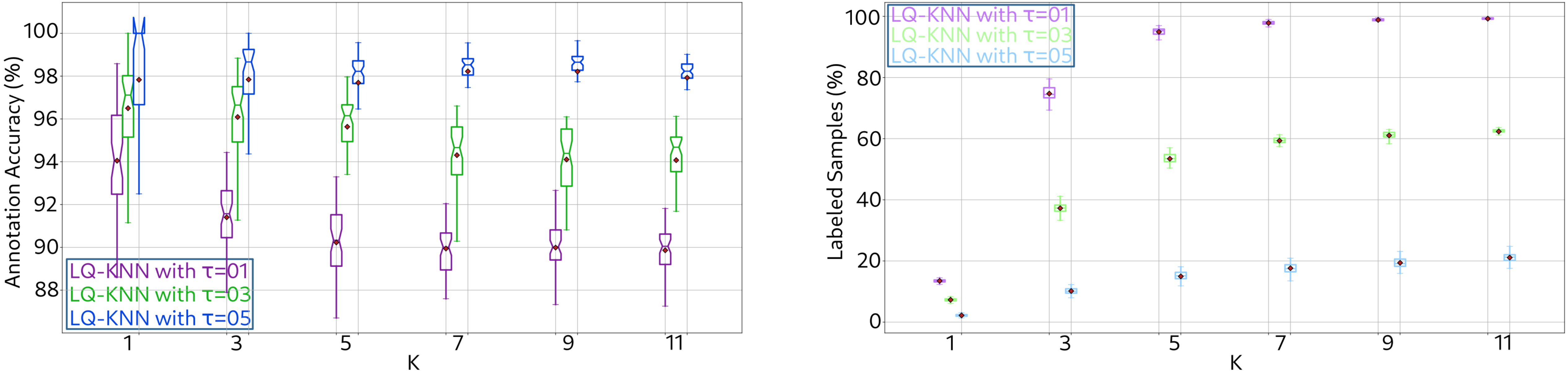


Figure - Comparison of *LQ-KNN* label propagation with different hyper-parameters using a HITS dataset

Proposed pipeline

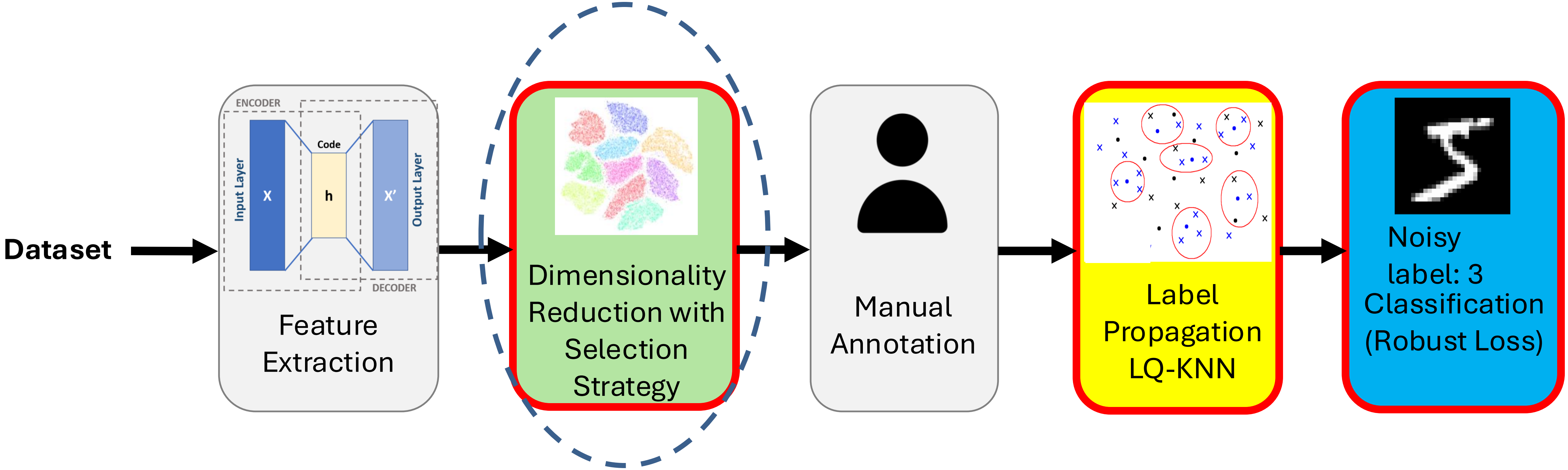


FIGURE - Semi-Automatic Data Annotation Method

Dimensionality Reduction

Silhouette Score[1]

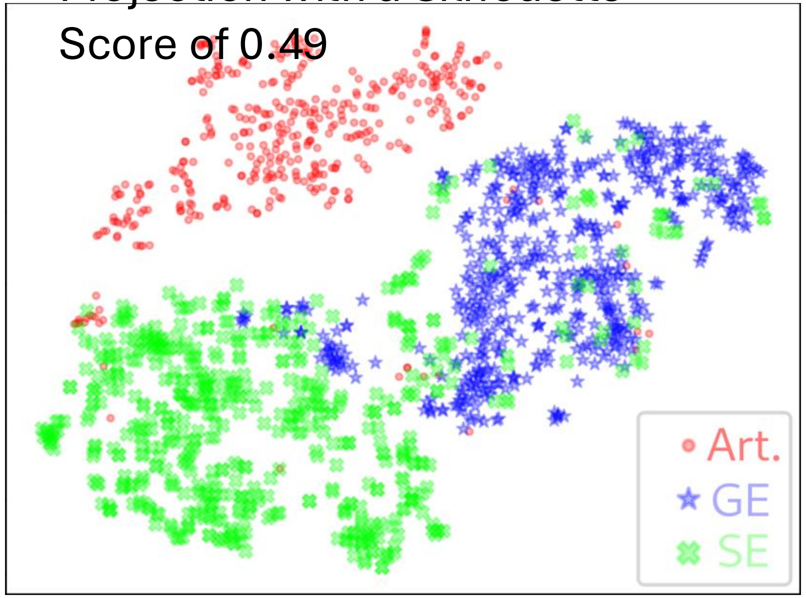
Compares the **similarity** of a sample k **between** :

- The **samples** of its **own class**.
- The samples of **other classes**.

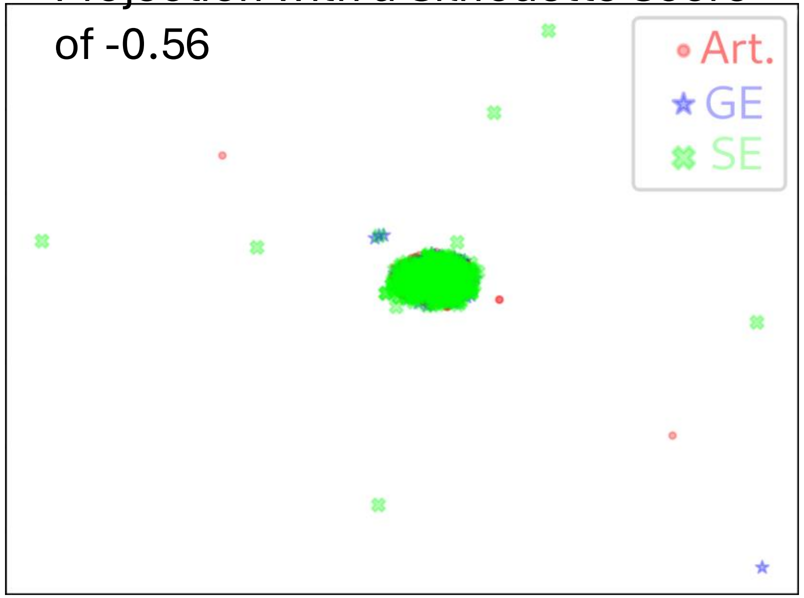
==> The higher the better

$$\forall k \in [1, L], s(k) = \begin{cases} \frac{\mu_{inter}(k) - \mu_{intra}(k)}{max(\mu_{inter}(k), \mu_{intra}(k))} & \text{if } |C_p| \geq 2 \\ 0 & \text{else} \end{cases}$$

Projection with a Silhouette Score of 0.49

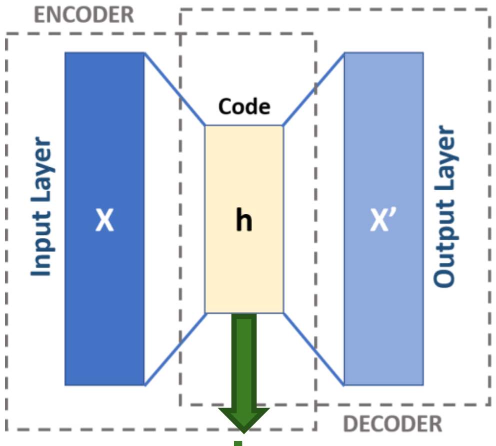


Projection with a Silhouette Score of -0.56



[1]Rousseeuw - 1987 - Silhouettes: A graphical aid to the interpretation and

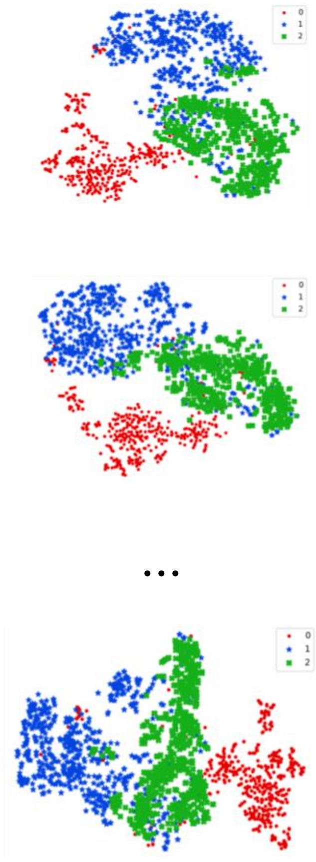
Dimensionality Reduction



S1	...	Sn
0.1		-0.9
-0.4		-1.7
...		...
-0.3		1.4
-1.2		-0.2

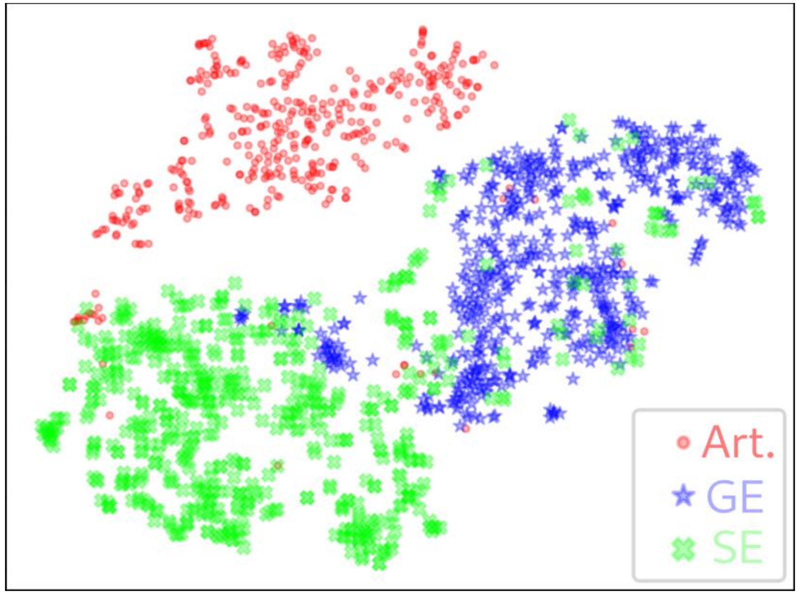
Features Dataset

t-SNE



Computed Projections
(3 t-SNE hyper-parameters)

Best Silhouette
Score Selection



Selected projection for manual annotation and label propagation

Contribution 1.a.: Semi-automatic data annotation method **state-of-the-art comparison**

Considered neighborhood to propagate the labels

Minimal local quality threshold

Dataset	Propagation method	$\|\mathcal{L}\|$	$\|\mathcal{U}\|$	τ	K	Annotation accuracy	Final % of labeled samples (%)	Annotation time (ms/sample)
MNIST	Std-KNN	1496	13504	-	5	91.83 ± 1.47	95.39 ± 1.05	$(30.98 \pm 5.84) \times 10^{-3}$
	Std-KNN	1496	13504	-	10	90.74 ± 1.45	99.43 ± 0.23	$(\mathbf{28.78 \pm 5.13}) \times \mathbf{10^{-3}}$
	LQ-KNN	1496	13504	0.1	5	**93.12 ± 1.36**	93.88 ± 0.66	$(59.10 \pm 12.35) \times 10^{-3}$
	LQ-KNN	1496	13504	0.1	10	92.66 ± 1.30	98.16 ± 0.42	$(50.48 \pm 11.32) \times 10^{-3}$
	OPF-semi	1496	13504	-	-	82.32 ± 6.17	**100.0 ± 0.0**	102.71 ± 17.52
OrganCMNIST	Std-KNN	1534	13858	-	5	81.87 ± 0.76	90.26 ± 2.64	$(26.33 \pm 2.65) \times 10^{-3}$
	Std-KNN	1534	13858	-	10	79.86 ± 0.67	99.00 ± 0.20	$(\mathbf{23.41 \pm 1.98}) \times \mathbf{10^{-3}}$
	LQ-KNN	1534	13858	0.1	5	**84.46 ± 0.57**	85.62 ± 1.99	$(53.00 \pm 7.47) \times 10^{-3}$
	LQ-KNN	1534	13858	0.1	10	82.73 ± 0.44	96.24 ± 1.09	$(44.36 \pm 5.69) \times 10^{-3}$
	OPF-semi	1534	13858	-	-	75.22 ± 4.48	**100.0 ± 0.0**	86.52 ± 0.51
HITS	Std-KNN	152	1393	-	5	82.12 ± 2.37	95.99 ± 1.70	$(10.39 \pm 0.20) \times 10^{-2}$
	Std-KNN	152	1393	-	10	81.36 ± 1.81	99.58 ± 0.63	$(\mathbf{10.04 \pm 0.18}) \times \mathbf{10^{-2}}$
	LQ-KNN	152	1393	0.1	5	**82.84 ± 2.12**	94.48 ± 1.72	$(16.87 \pm 0.48) \times 10^{-3}$
	LQ-KNN	152	1393	0.1	10	82.67 ± 2.02	98.50 ± 0.80	$(16.13 \pm 0.35) \times 10^{-2}$
	OPF-semi	152	1393	-	-	78.40 ± 13.44	**100.0 ± 0.0**	9.48 ± 1.1

Table – Label propagation methods comparison on different datasets

Metrics:

Annotation accuracy:

$$\frac{\text{\# correct new labeled samples}}{\text{\# new labeled samples}}$$

Percentage of new labeled samples:

$$\frac{\text{\# new labeled samples}}{\text{\# originally unlabeled samples}}$$

Contribution 1.b: Optimal 2D projection selection.

Dataset: HITS Dataset.

Evaluation: Label Propagation on 2 different projections.

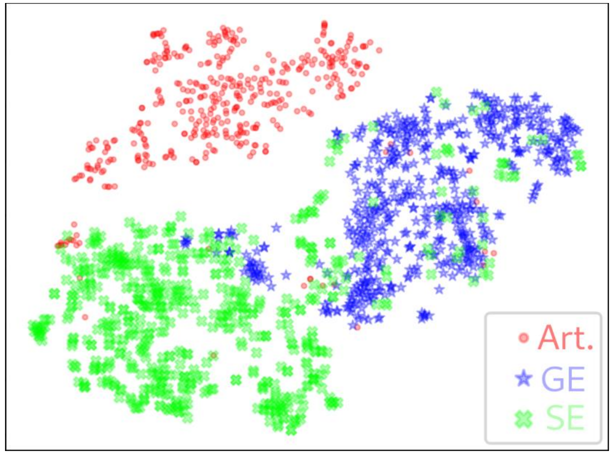
Metrics: Annotation accuracy and percentage of new labeled samples.

Results:

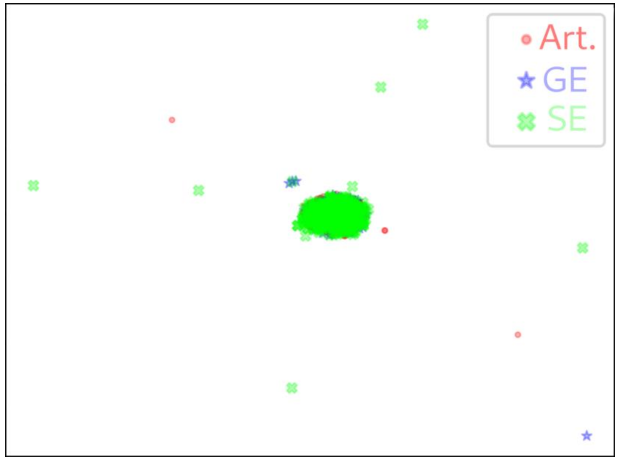
K	Propagation method	Projection	$\|\mathcal{L}\|$	$\|\mathcal{U}\|$	τ	Annotation accuracy	Final % of labeled samples
5	**Std-KNN**	**Best**	**152**	**1393**	**-**	$\mathbf{89.8 \pm 1.63}$	$\mathbf{95.52 \pm 1.23}$
	Std-KNN	Worst	152	1393	-	52.2 ± 2.53	98.78 ± 0.42
5	**LQ-KNN**	**Best**	**152**	**1393**	**0.1**	$\mathbf{90.23 \pm 1.46}$	$\mathbf{94.93 \pm 1.32}$
	LQ-KNN	Worst	152	1393	0.1	70.69 ± 2.64	57.4 ± 2.01

Conclusion:
- Projection selection improves annotation accuracy.
- Our proposed method is more robust against bad projections.

Best Projection



Worst Projection



Contribution 1.b: Optimal 2D projection selection.

Dataset: HITS Dataset.

Evaluation: Label Propagation on 2 different projections.

Metrics: Annotation accuracy and percentage of new labeled samples.

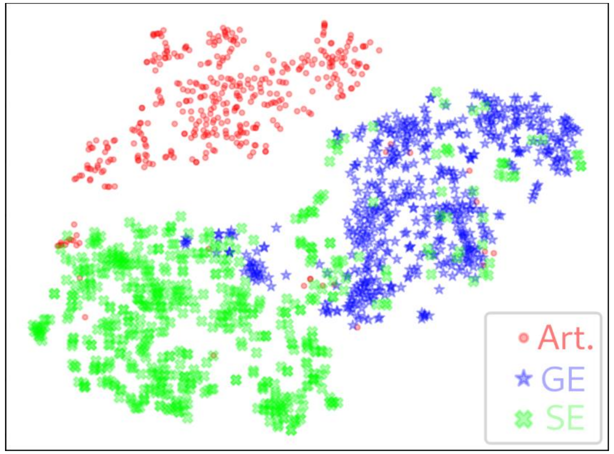
Results:

K	Propagation method	Projection	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{U}\rvert$	τ	Annotation accuracy	Final % of labeled samples
5	**Std-KNN**	**Best**	**152**	**1393**	**-**	$\mathbf{89.8 \pm 1.63}$	$\mathbf{95.52 \pm 1.23}$
	Std-KNN	Worst	152	1393	-	52.2 ± 2.53	98.78 ± 0.42
5	**LQ-KNN**	**Best**	**152**	**1393**	**0.1**	$\mathbf{90.23 \pm 1.46}$	$\mathbf{94.93 \pm 1.32}$
	LQ-KNN	Worst	152	1393	0.1	70.69 ± 2.64	57.4 ± 2.01

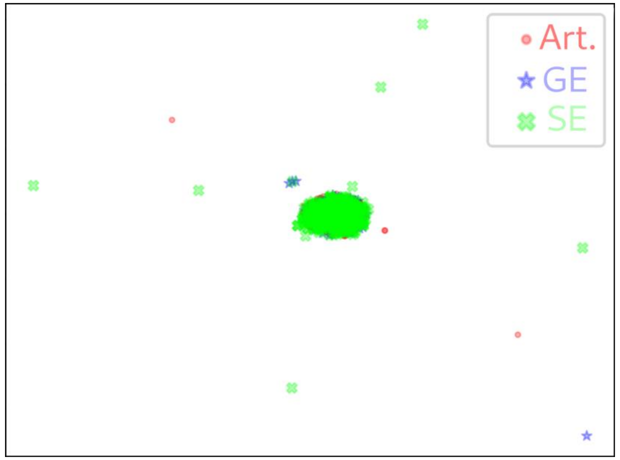
Conclusion:

- Projection selection improves annotation accuracy.
- Our proposed method is more robust against bad projections.

Best Projection



Worst Projection



Contribution 1.b: Optimal 2D projection selection.

Dataset: HITS Dataset.

Evaluation: Label Propagation on 2 different projections.

Metrics: Annotation accuracy and percentage of new labeled samples.

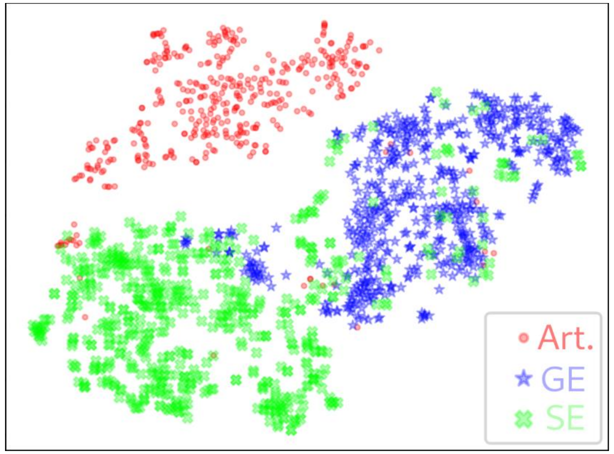
Results:

K	Propagation method	Projection	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{U}\rvert$	τ	Annotation accuracy	Final % of labeled samples
5	**Std-KNN**	**Best**	**152**	**1393**	**-**	$\mathbf{89.8 \pm 1.63}$	$\mathbf{95.52 \pm 1.23}$
	Std-KNN	Worst	152	1393	-	52.2 ± 2.53	98.78 ± 0.42
5	**LQ-KNN**	**Best**	**152**	**1393**	**0.1**	$\mathbf{90.23 \pm 1.46}$	$\mathbf{94.93 \pm 1.32}$
	LQ-KNN	Worst	152	1393	0.1	70.69 ± 2.64	57.4 ± 2.01

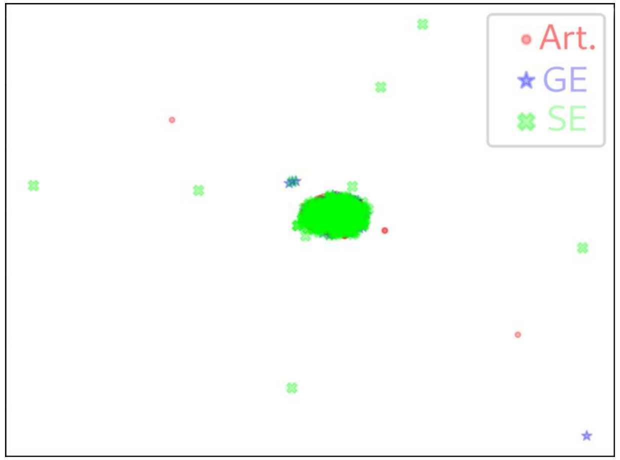
Conclusion:

- Projection selection improves annotation accuracy.
- Our proposed method is more robust against bad projections.

Best Projection



Worst Projection



Contribution 1.c: Classification using robust loss functions to compensate the noise in the labels.

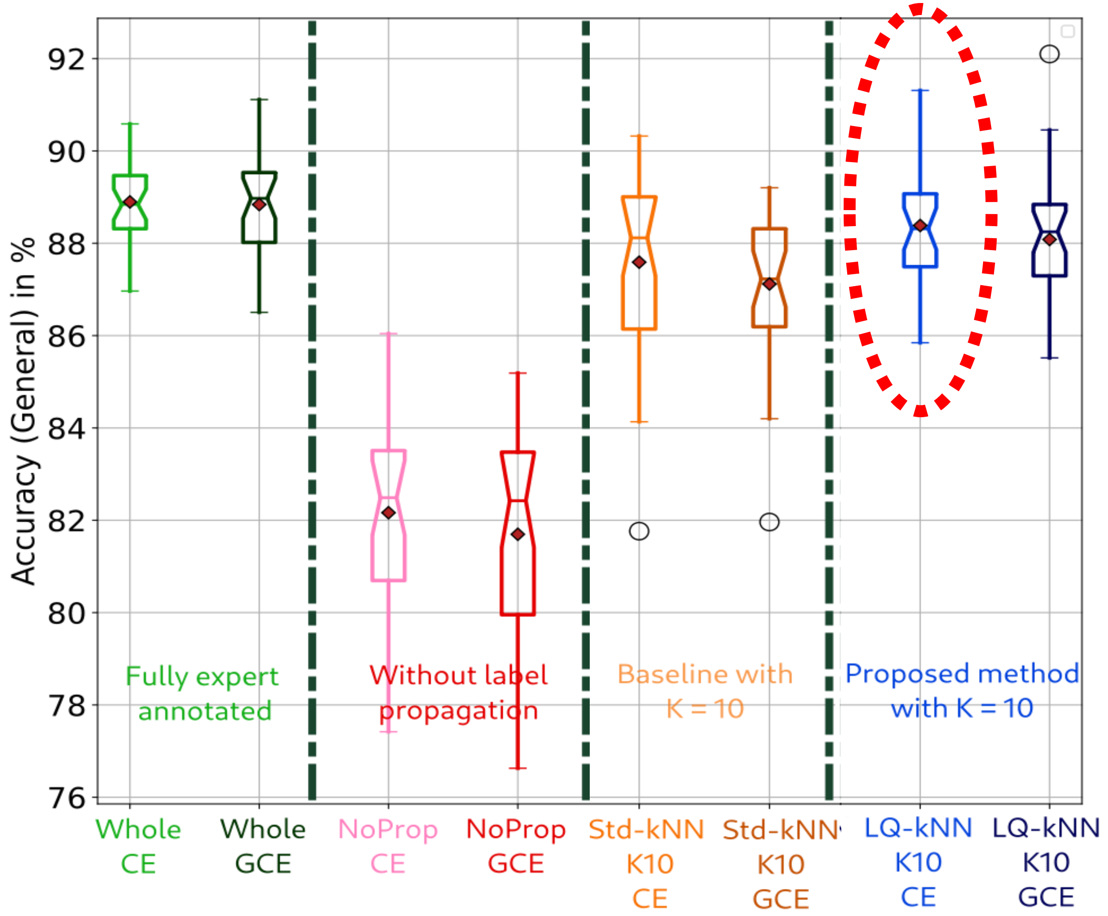
Dataset

Dataset	Core Dataset	Prop. method	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{U}\rvert$	# of automatically labeled samples	Mean annot. accuracy	K	τ
HITS No Prop.	HITS	No Prop.	152	1393	-	-	-	-
HITS Whole		No Prop.	1545	0	-	-	-	-
HITS Std-KNN-K10		Std-KNN			1390 ± 2	88.72 ± 2.33	10	-
HITS LQ-KNN-K10		LQ-KNN			1382 ± 3	89.92 ± 1.42	10	0.1

Metrics:

Classification accuracy.
Classification class accuracy.

==> Our method allows to increase the classification accuracy by 6 % with respect to using a reduced dataset (no propagation)



HITS Datasets Results

Database Creation

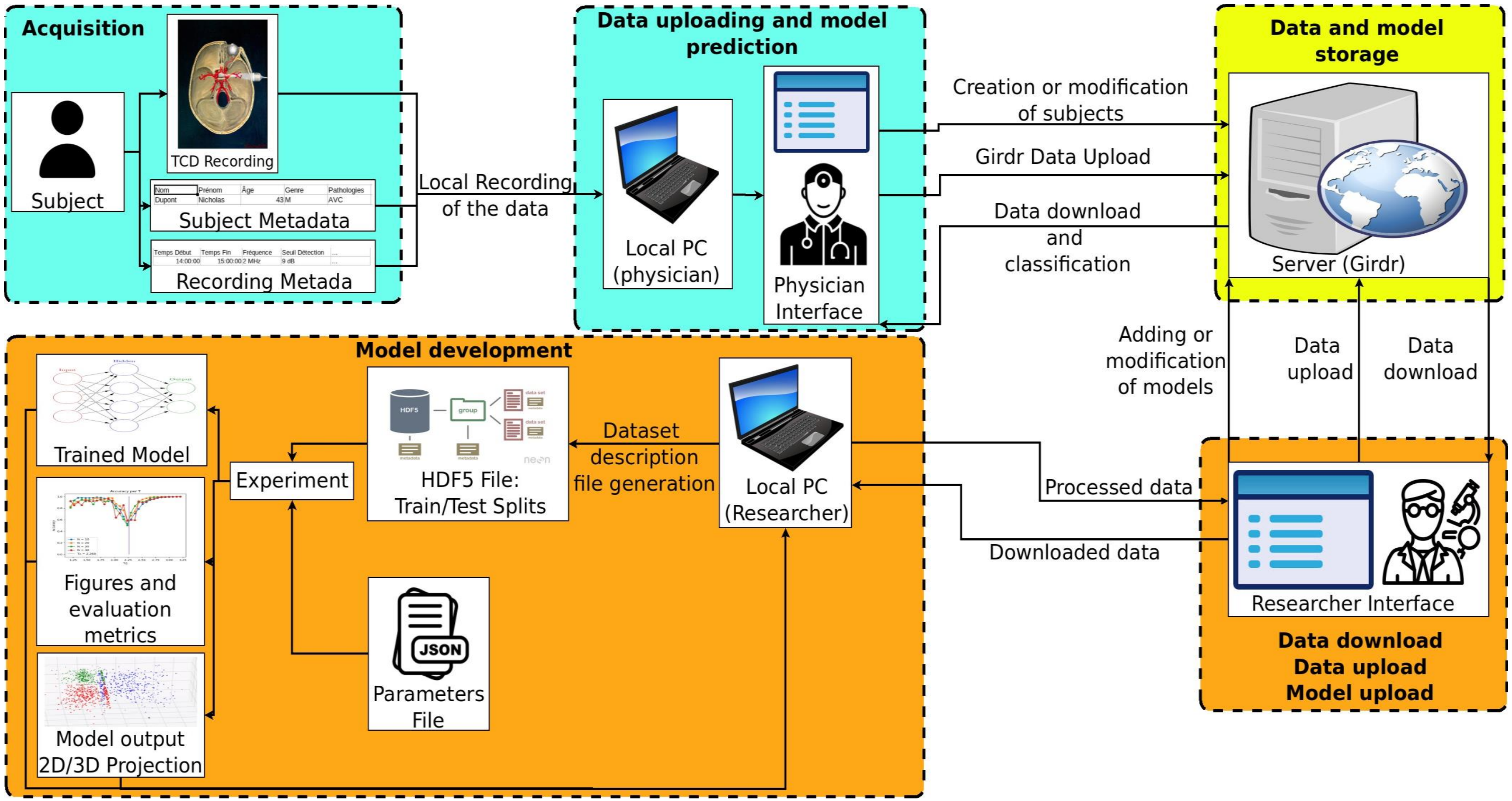


Figure – Data pipeline. Two types of data : raw and derivative.

Database Structure

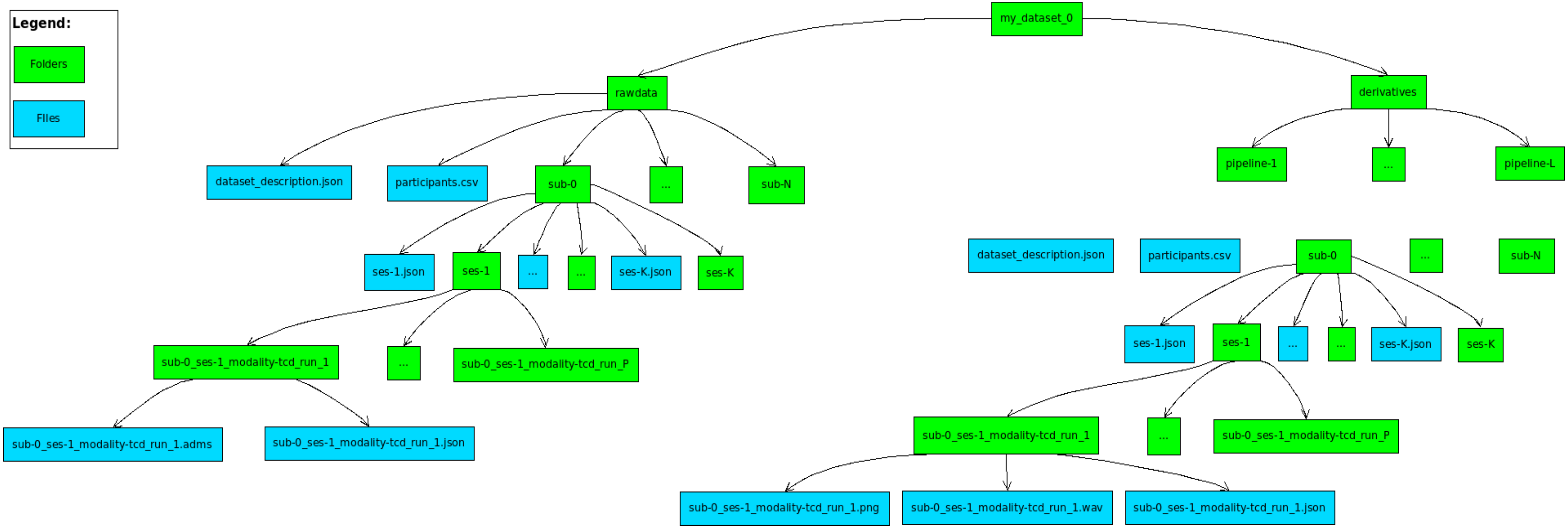


Figure – Database structure on Gridr

Auto-encoders architectures

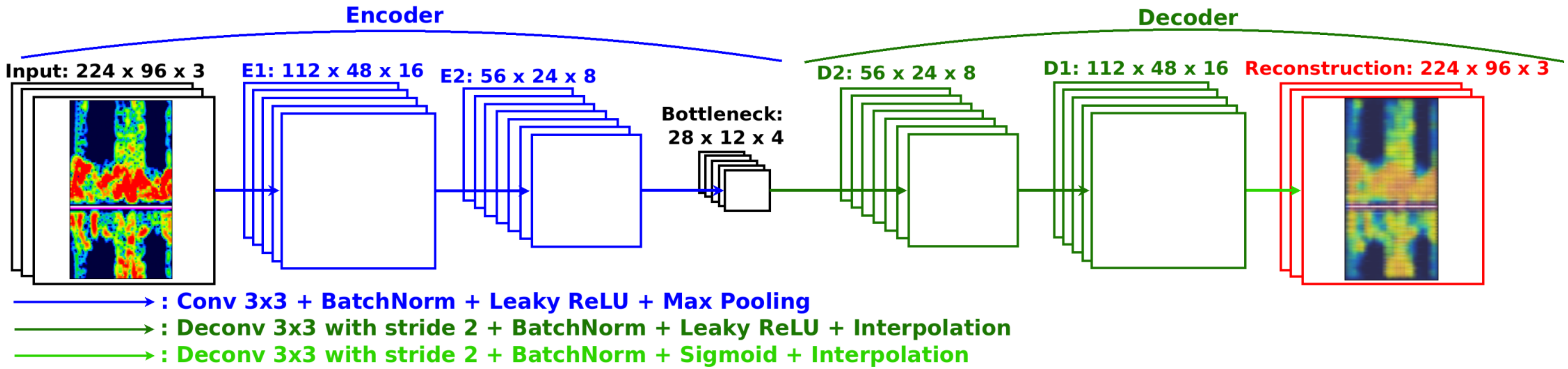


Figure – Convolutional auto-encoder for the HITS dataset.

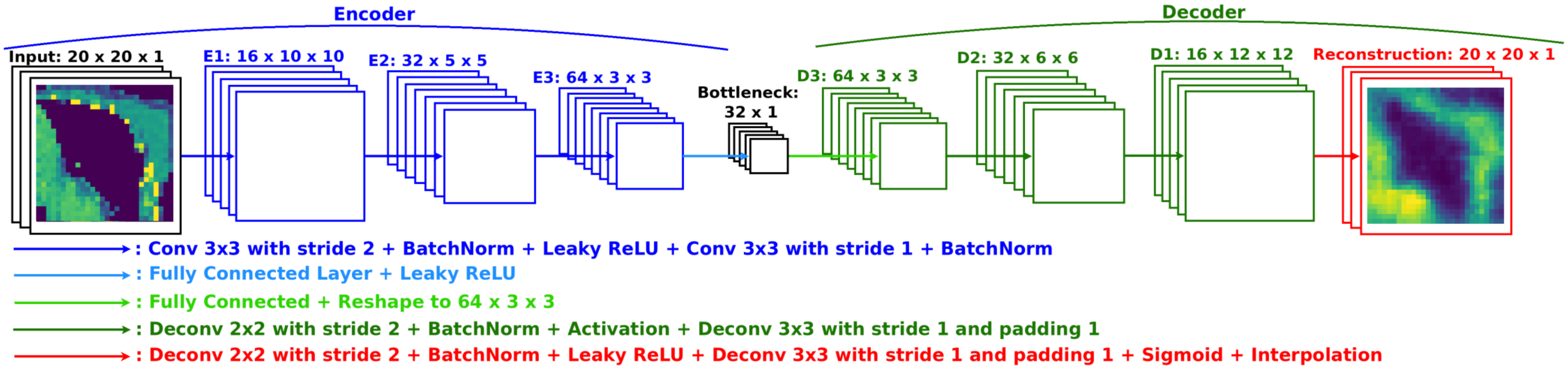


Figure – Convolutional auto-encoder for the OrganCMNIST and MNIST datasets.

Classifiers architectures

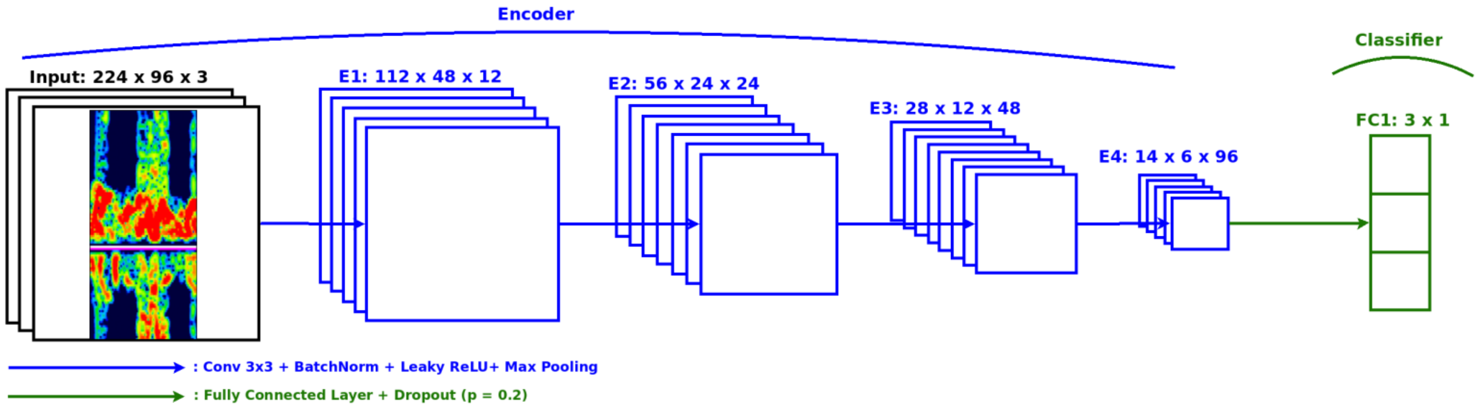


Figure – Convolutional classifier for the HITS dataset.

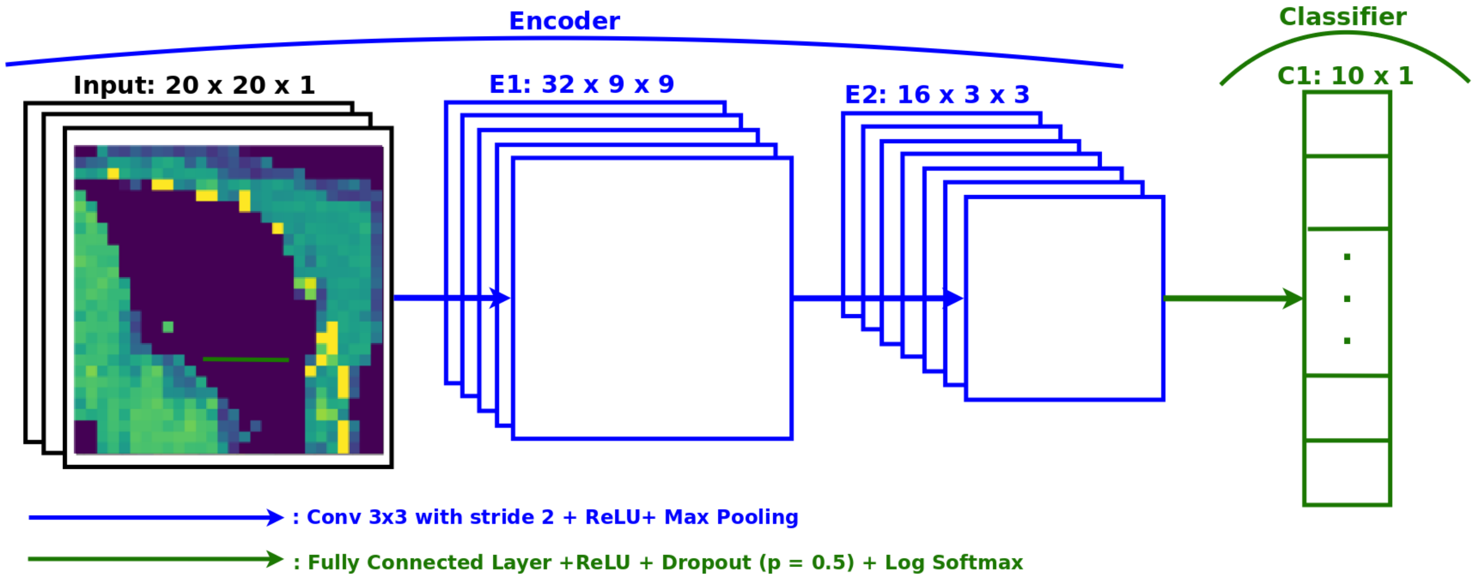


Figure – Convolutional classifier for the OrganCMNIST and MNIST datasets.

Semi-automatic data annotation

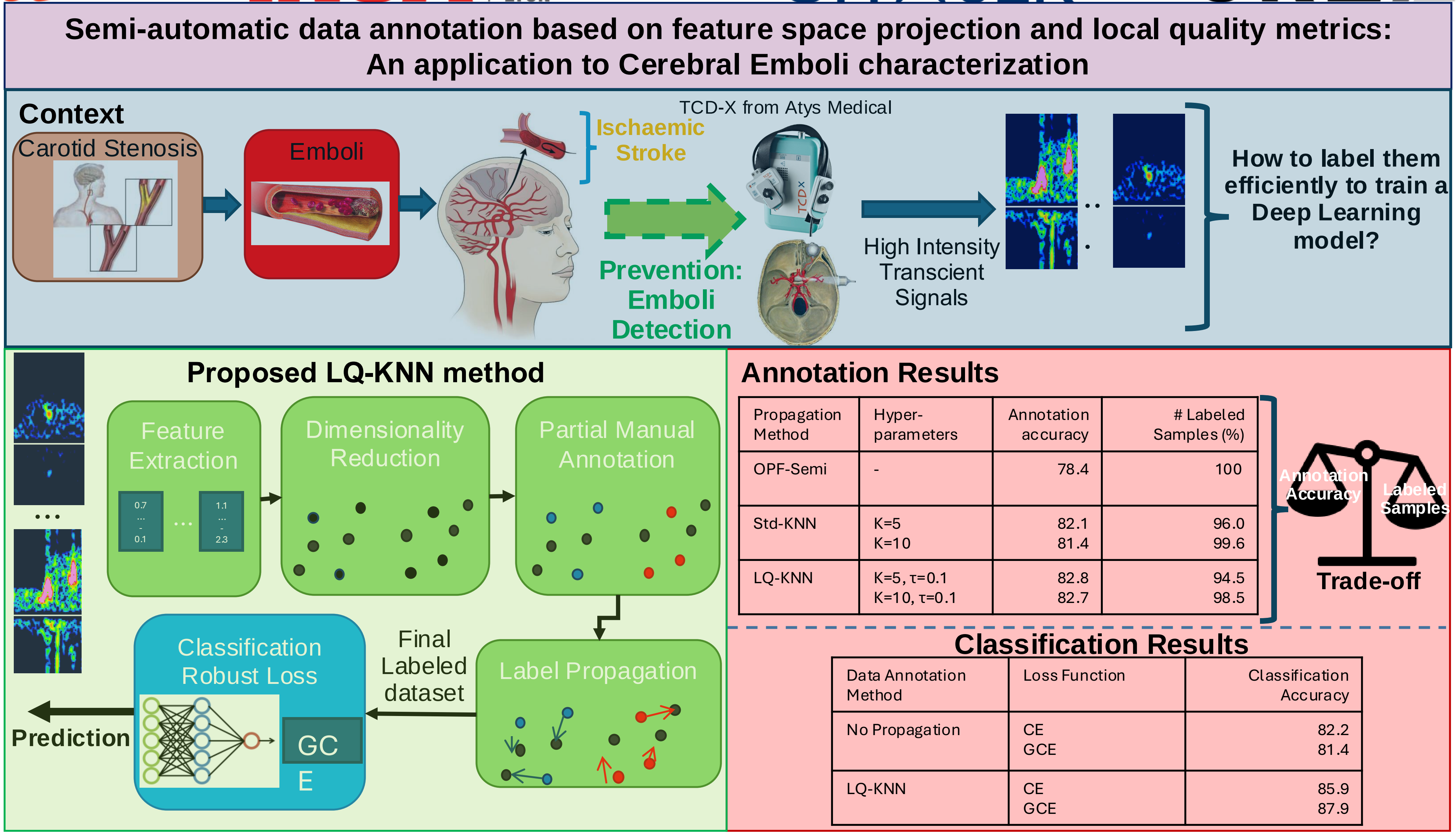


FIGURE - Graphical abstract of Vindas et al. (2022) in Medical Image Analysis

Contribution 2 : Multi-feature medical signal classification

ECG 1D CNN-transformer

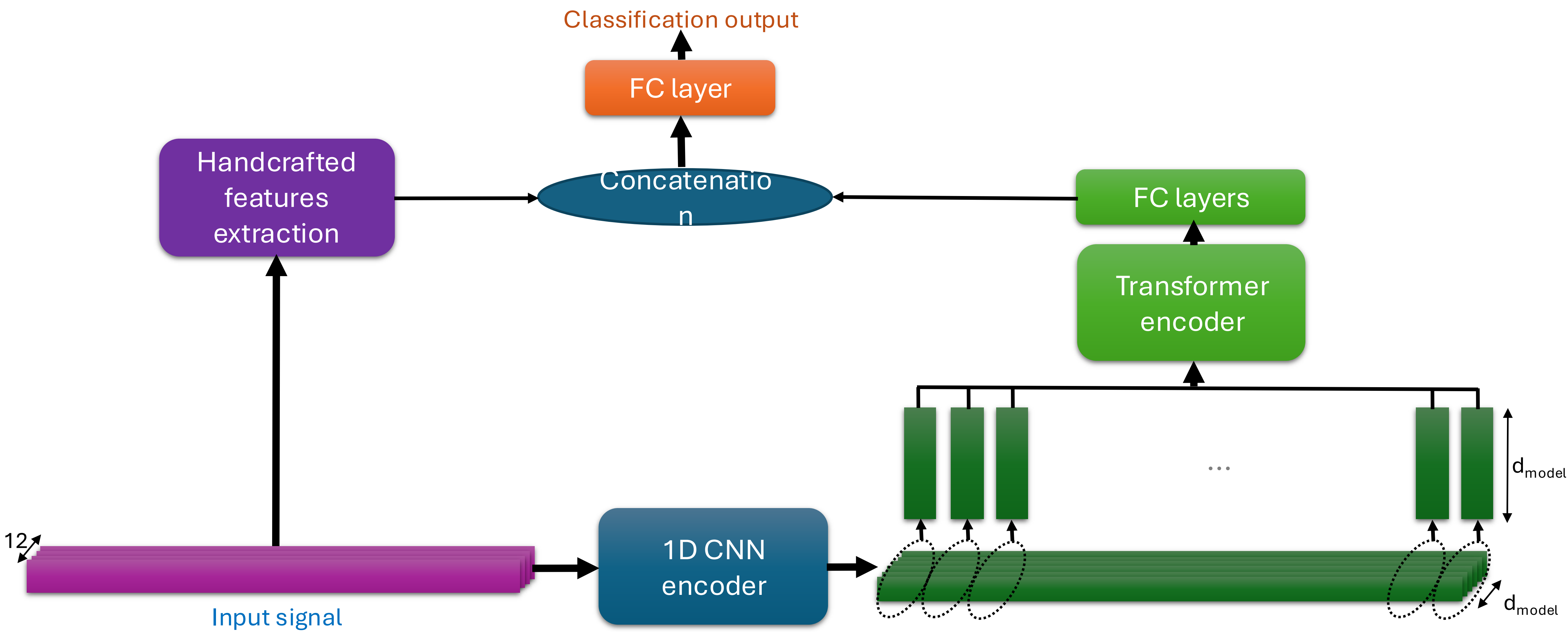


Figure - Proposed 1D CNN-transformer model for ECG signal classification (**Natarajan et al., 2020.**)

Link constraints regularization

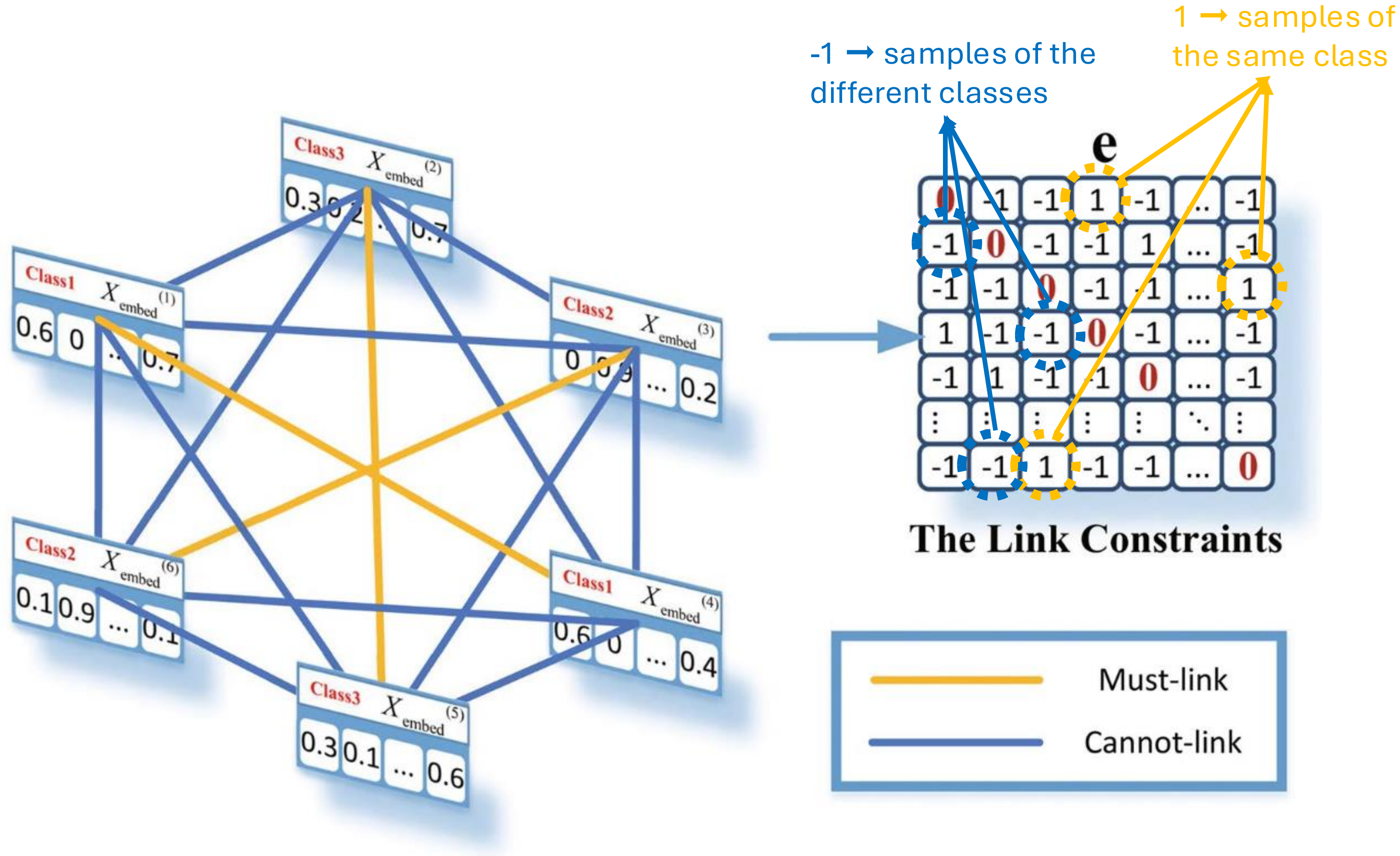
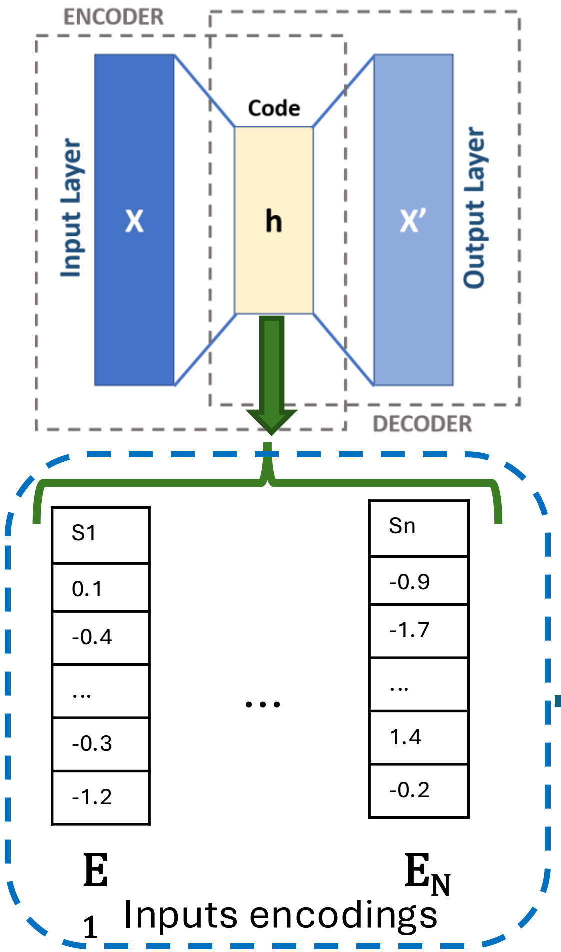


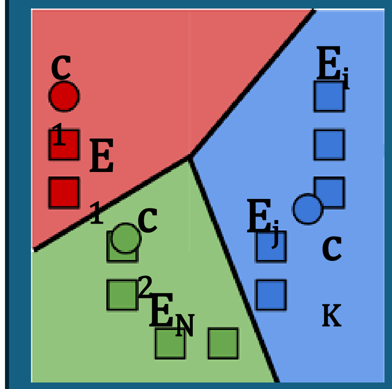
Figure – Links constraints regularization illustration for transformer models (**Che et al., 2021**)

Deep embedded clustering (**DEC**)



Clustering

Initialization



K-means

Soft assignments of E_i to C_j

$$q_{ij} = \frac{\sum_{p=1}^{J}\left(1+\frac{||\mathcal{E}(X_i)-\mathbf{c}_p||^2}{\alpha}\right)^{\frac{\alpha+1}{2}}}{\left(1+\frac{||\mathcal{E}(X_i)-\mathbf{c}_j||^2}{\alpha}\right)^{\frac{\alpha+1}{2}}}$$

Clusters soft frequencies

$$f_j = \sum_{p=1}^{J} q_{pj}$$

Target distribution

$$p_{ij} = \frac{\frac{q_{ij}^2}{f_j}}{\sum_{p=1}^{J}\frac{q_{ip}^2}{f_p}}$$

Properties

- Cluster purity.
- Focus on high confidence samples.
- Normalization of the contribution of each centroid to the final loss.

Deep embedded clustering (DEC) loss function

$$\mathcal{L}_{DEC} = KL(P||Q)$$

Figure – Deep embedded clustering (**DEC**) for unsupervised learning (**Xie et al., 2016**)

Single feature TFR 2D CNN

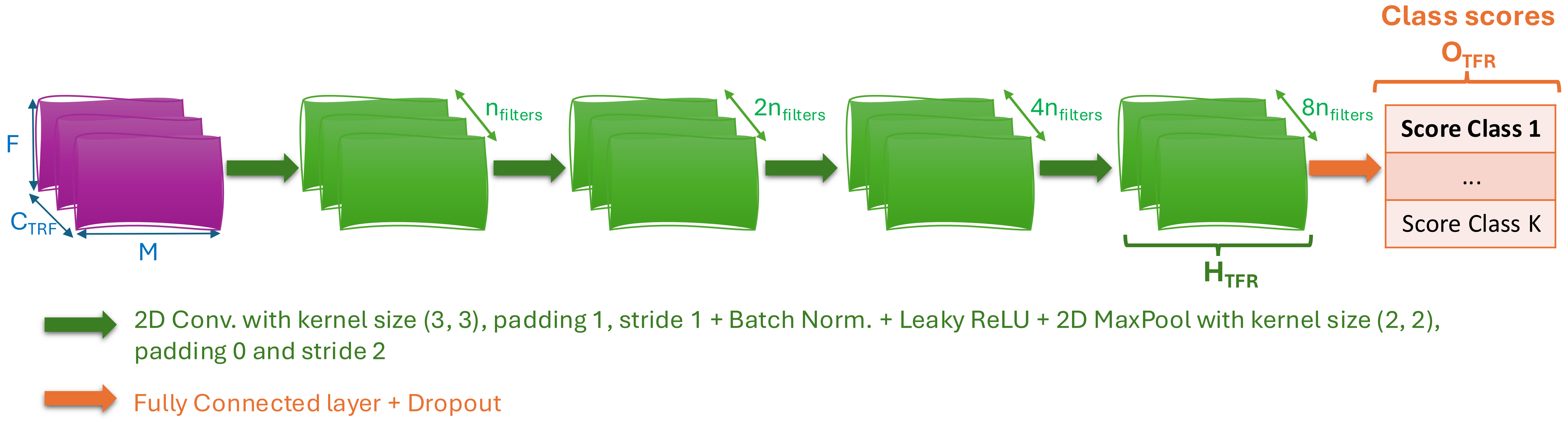


Figure - Proposed 2D CNN

Single feature raw signal 1D CNN-transformer

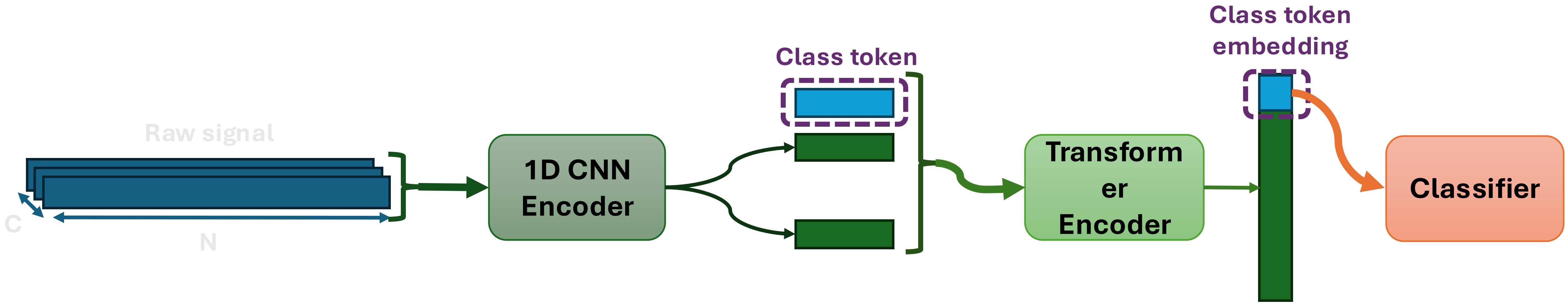


Figure - Proposed 1D CNN Transformer architecture (inspired from Natarajan et al. 2020).

Guided and regularized intermediate fusion

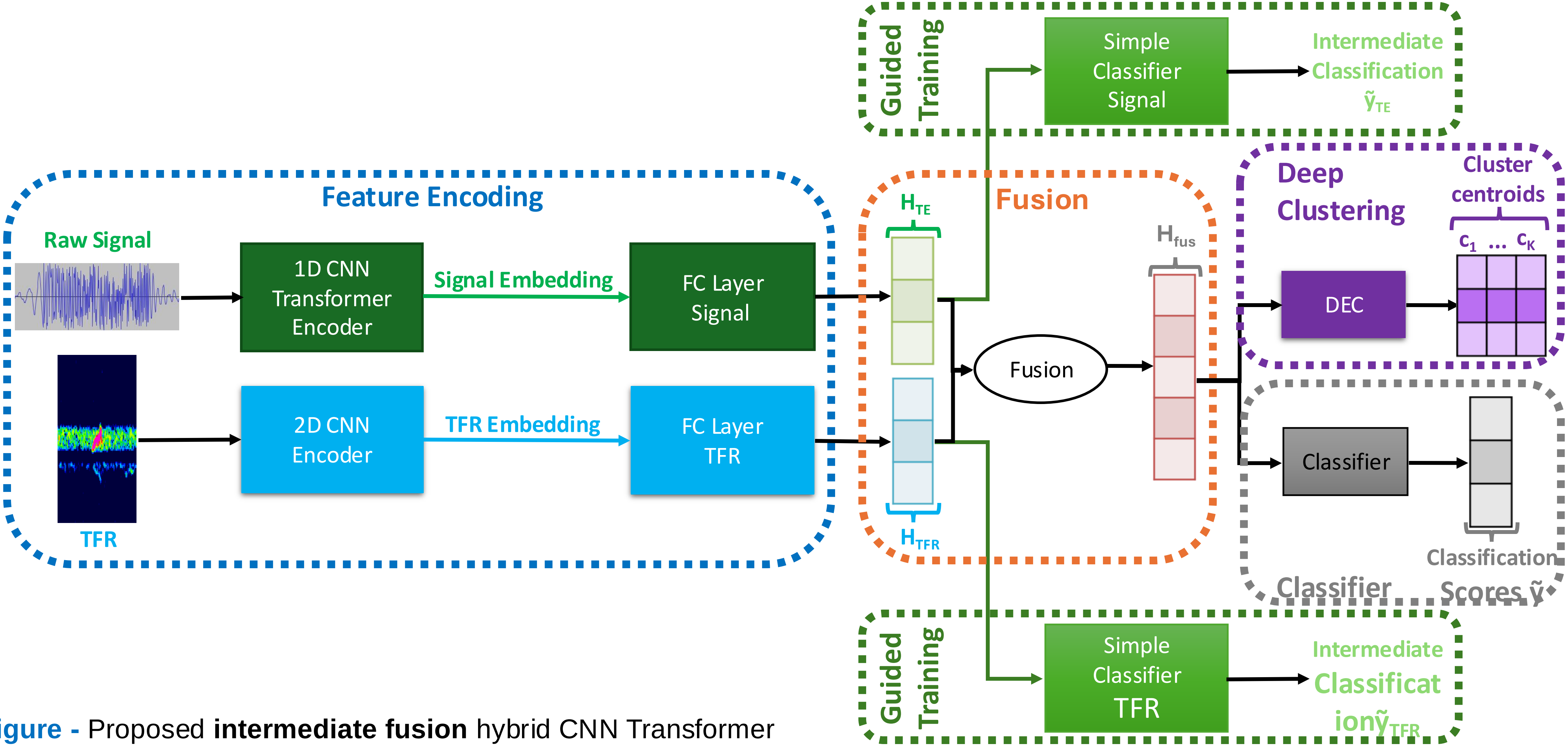


Figure - Proposed **intermediate fusion** hybrid CNN Transformer model.

Intermediate fusion

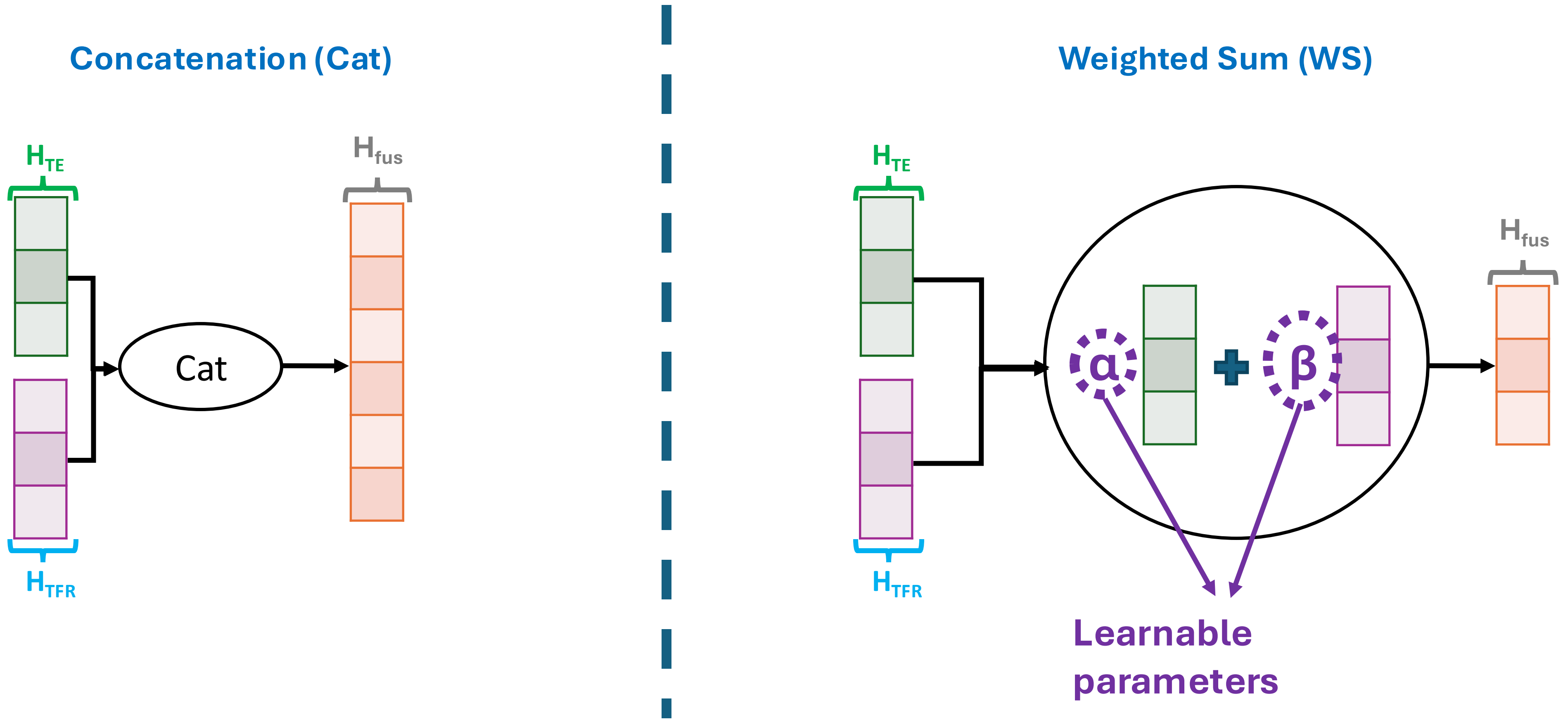


Figure - Proposed intermediate fusion strategies: concatenation and weighted sum

Single feature raw signal 1D CNN-transformer

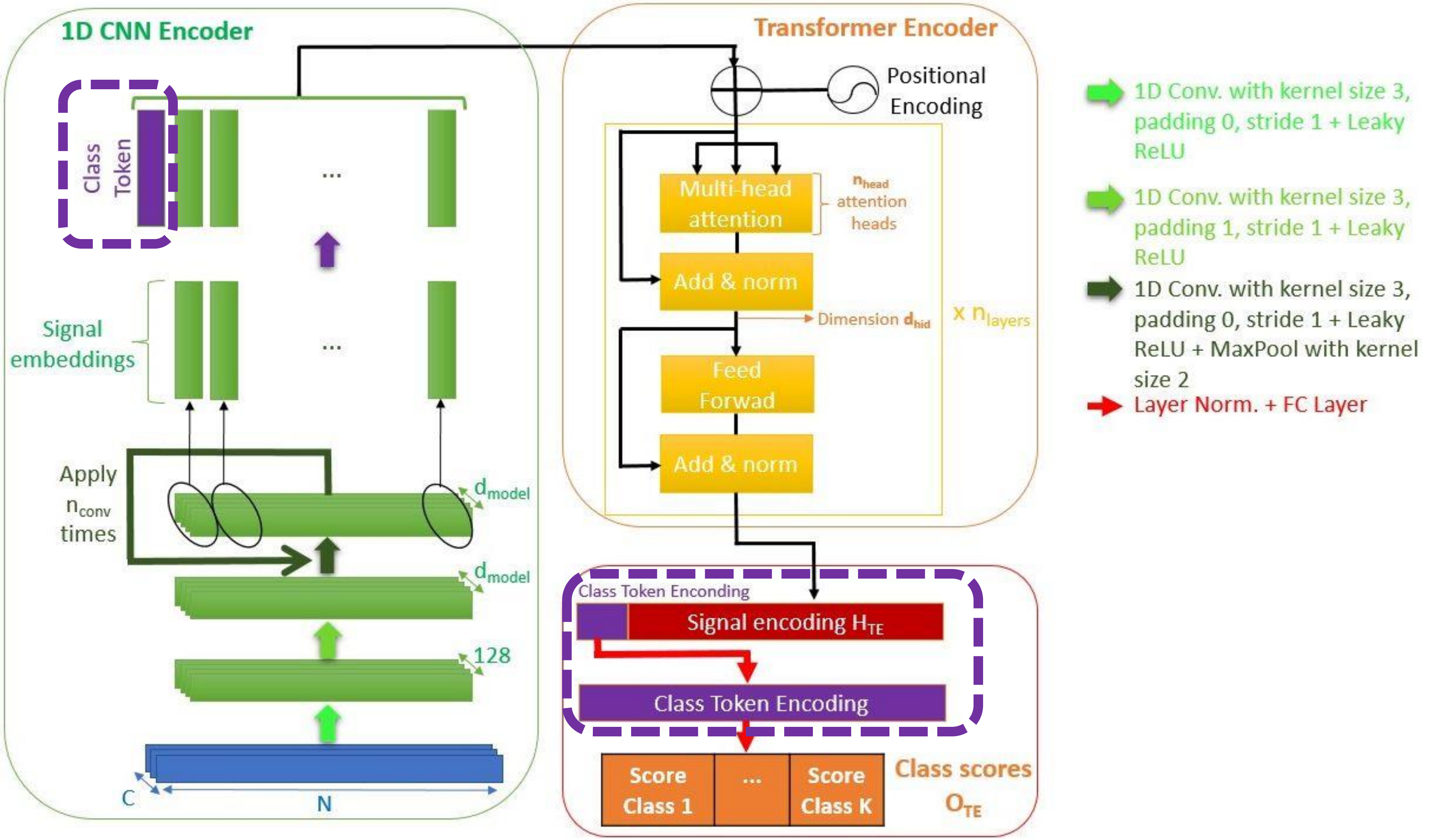
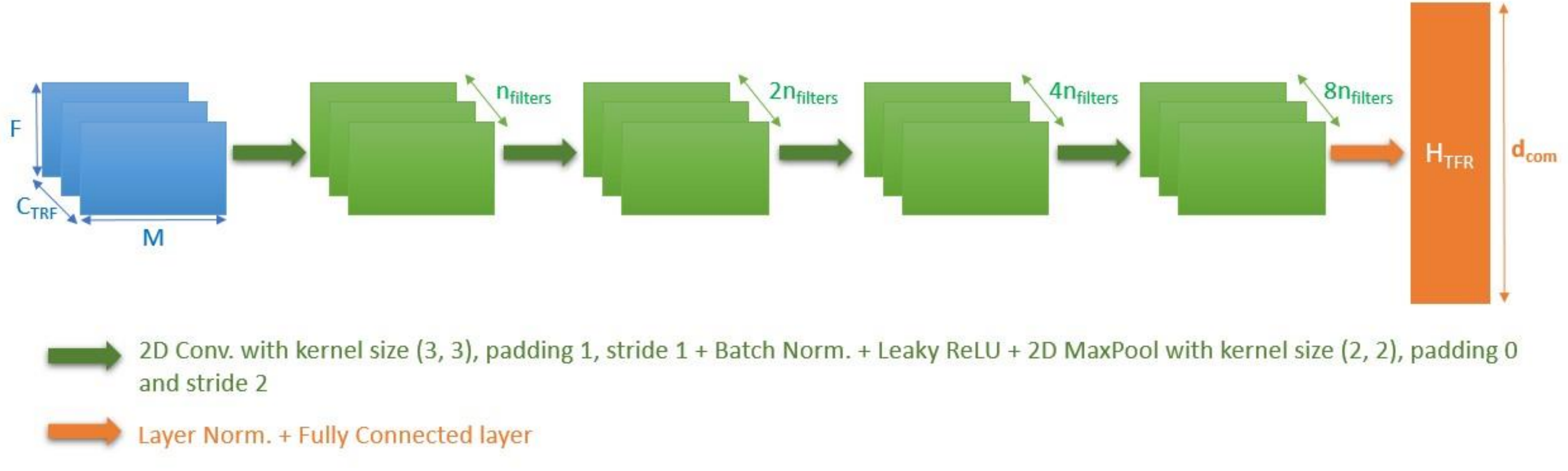
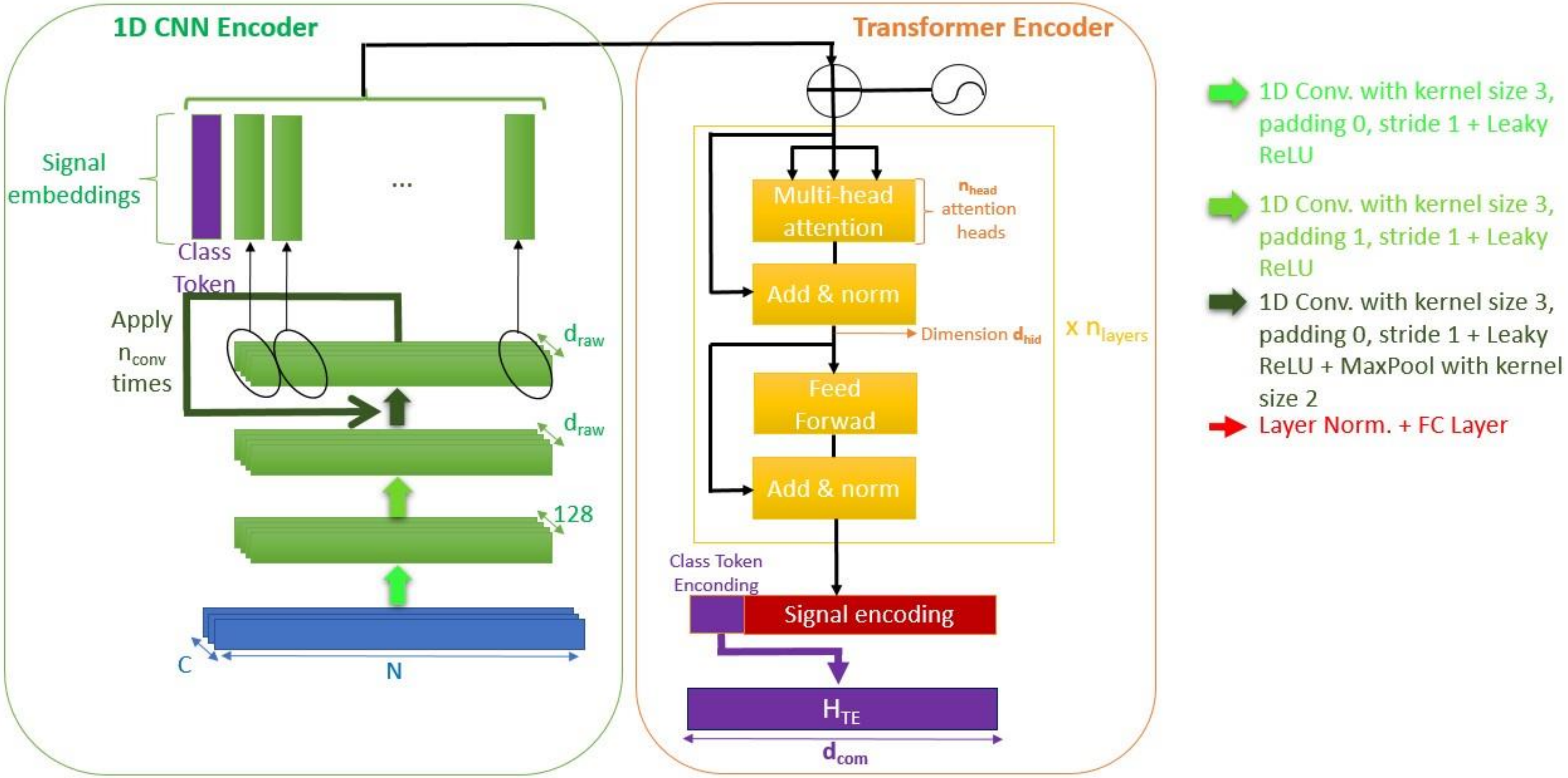


Figure - Proposed 1D CNN Transformer architecture (inspired from Natarajan et al. 2020).

CNN model late fusion approach

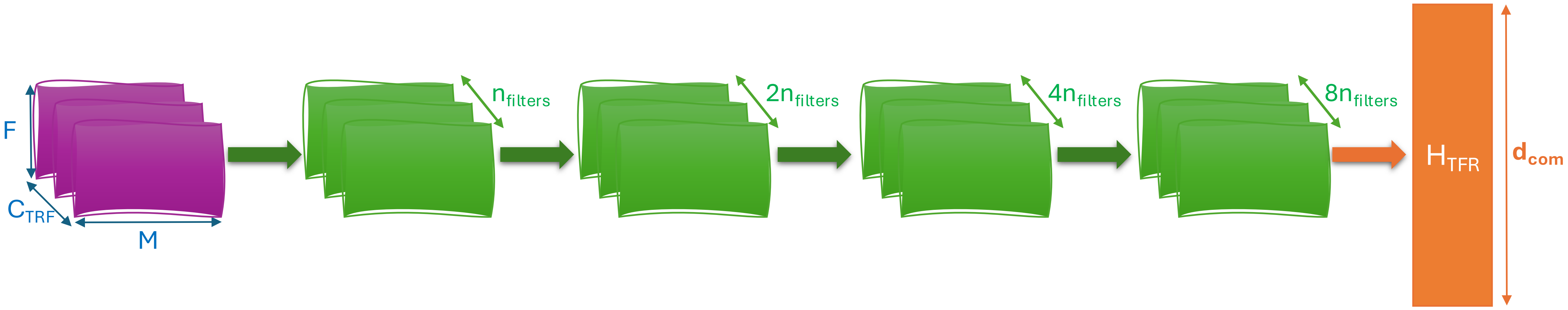


Transformer model intermediate fusion approach



Inspired from Natarajan et al. (2020), A wide and deep transformer neural network for 12-lead ecg classification.

CNN model intermediate fusion approach



2D Conv. with kernel size (3, 3), padding 1, stride 1 + Batch Norm. + Leaky ReLU + 2D MaxPool with kernel size (2, 2), padding 0 and stride 2

Layer Norm. + Fully Connected layer

Late fusion attention weights

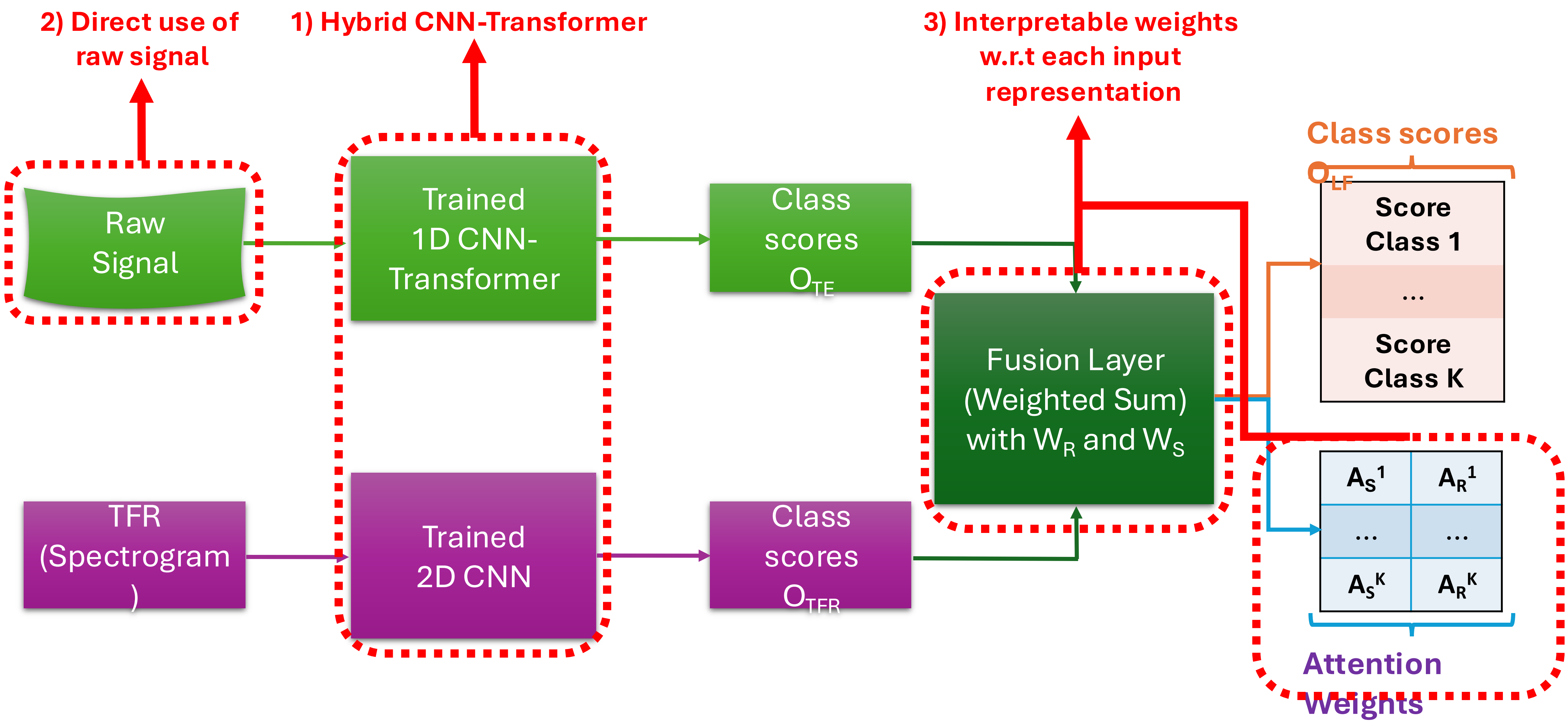


Figure - Proposed hybrid CNN Transformer global model

Guided and regularized intermediate fusion

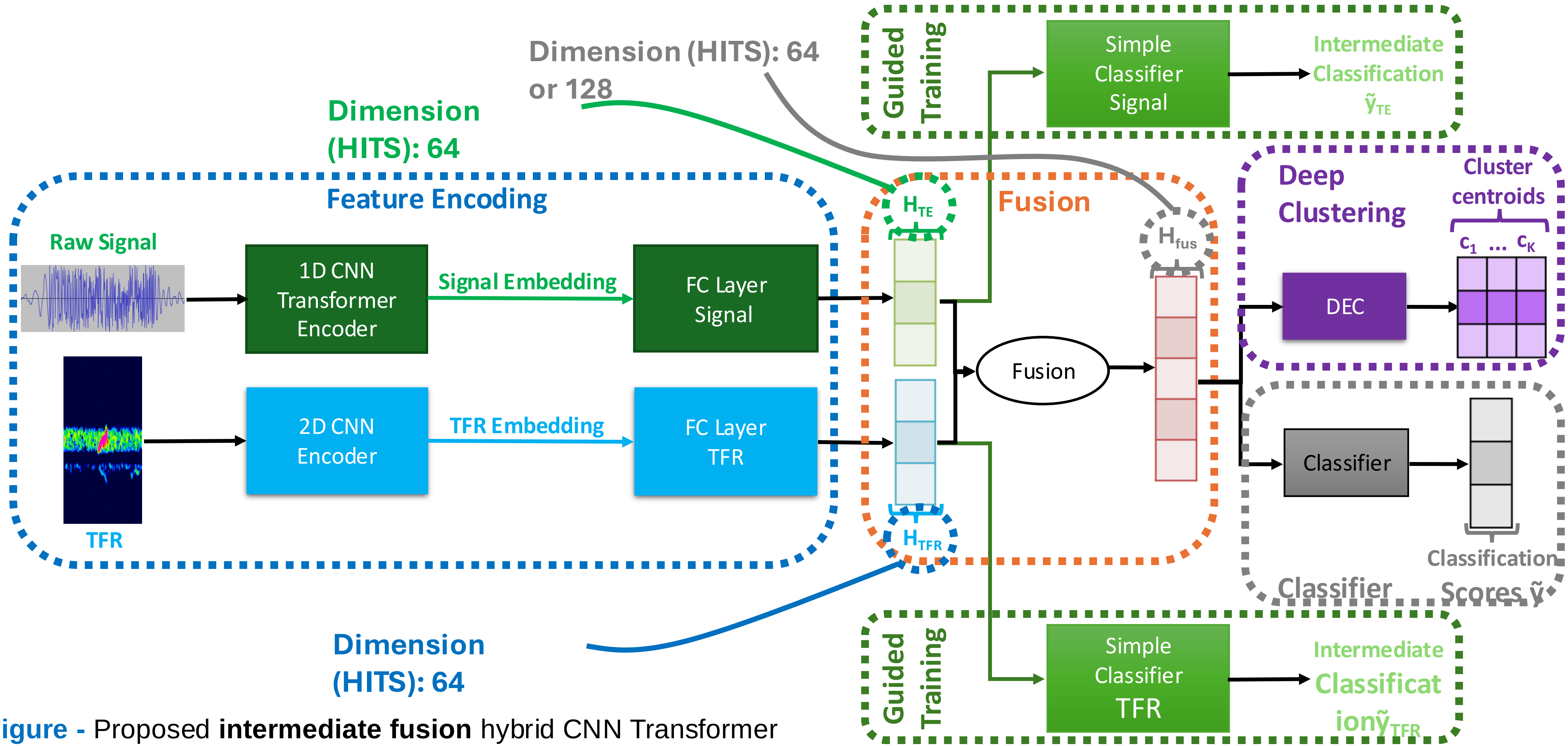


Figure - Proposed **intermediate fusion** hybrid CNN Transformer model.

Results: SOTA comparison HITS validation

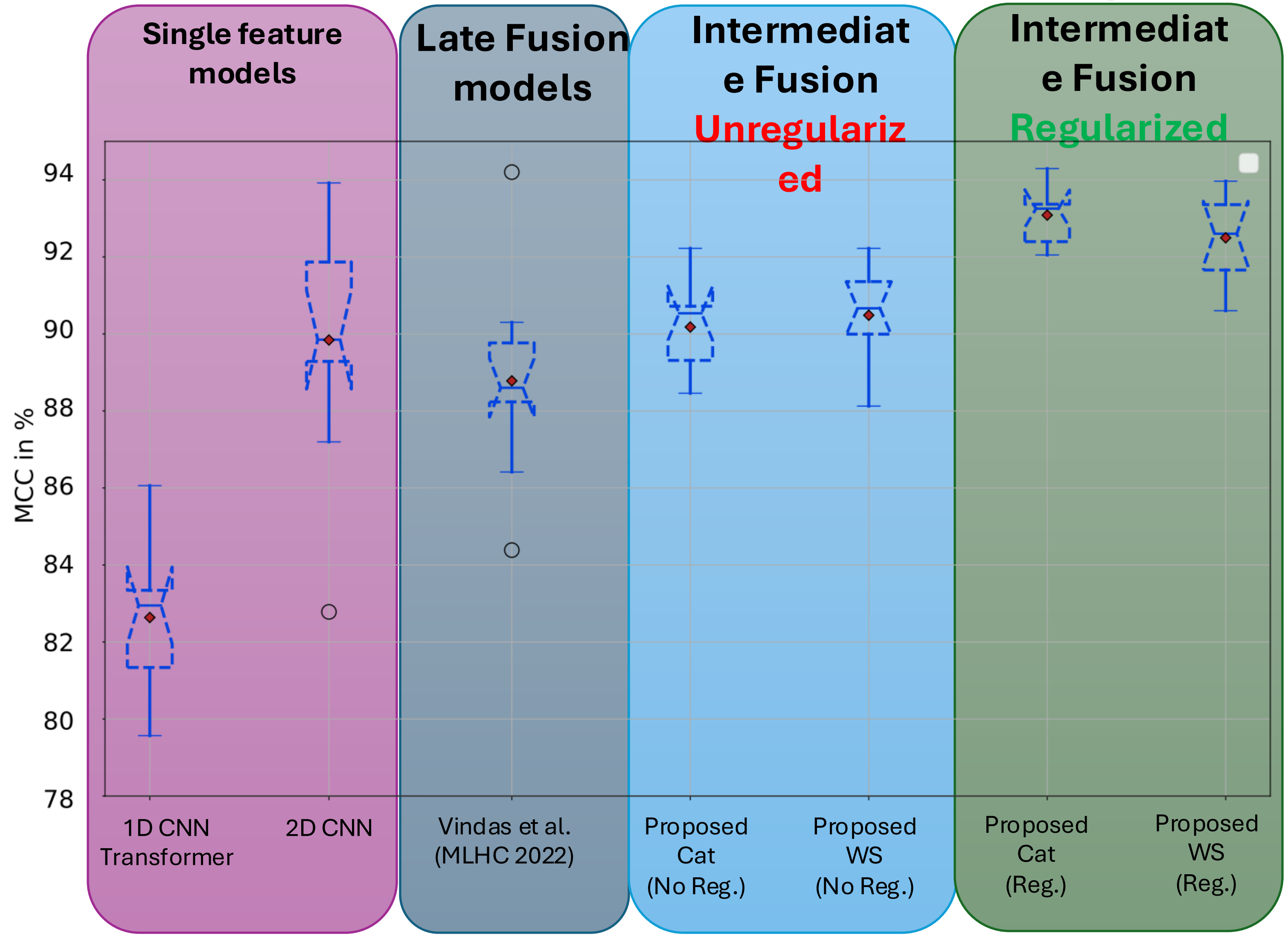
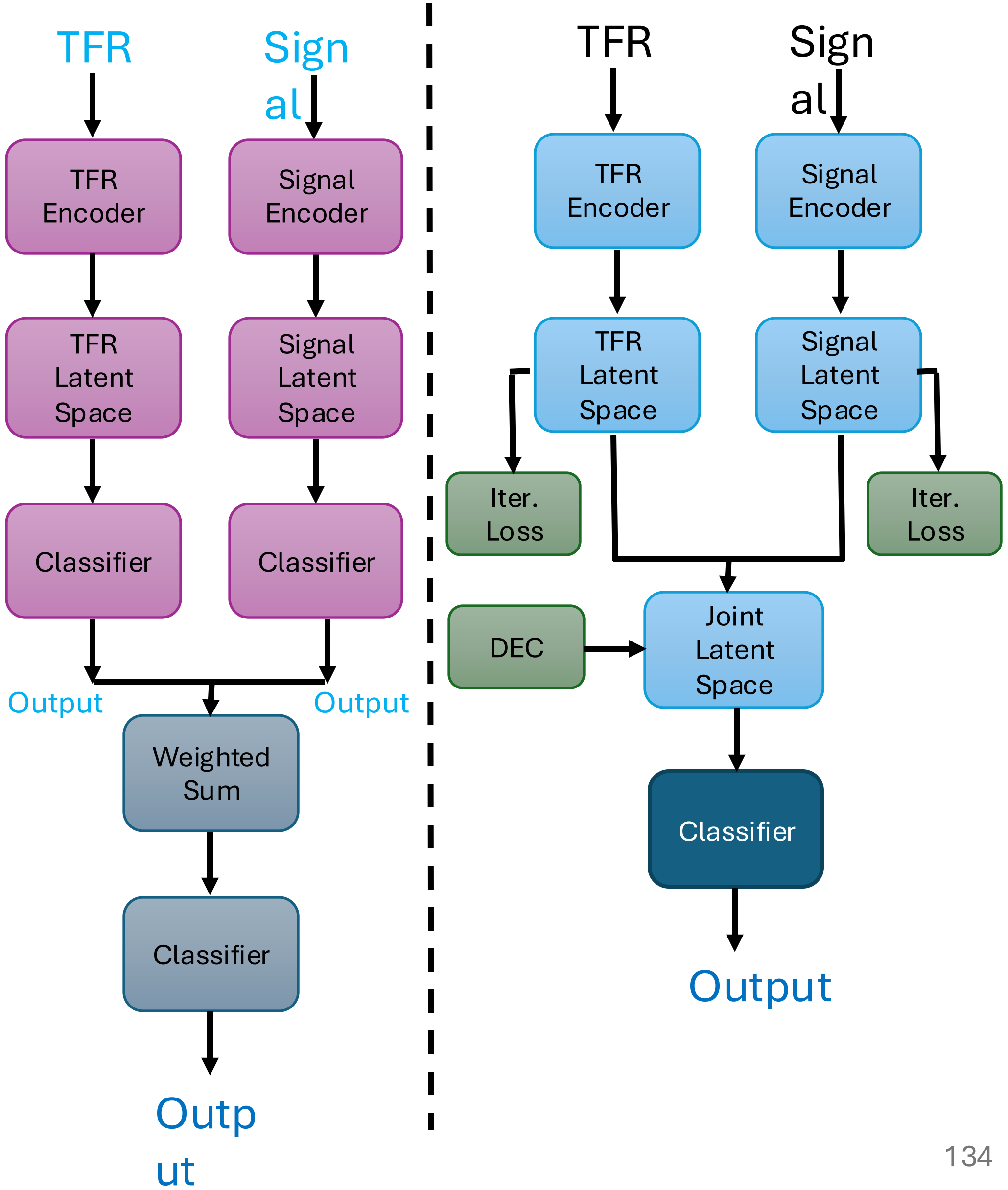


Figure - Comparison of the classification performances of different single and multi-feature models on the HITS dataset



Results Multi-feature classification

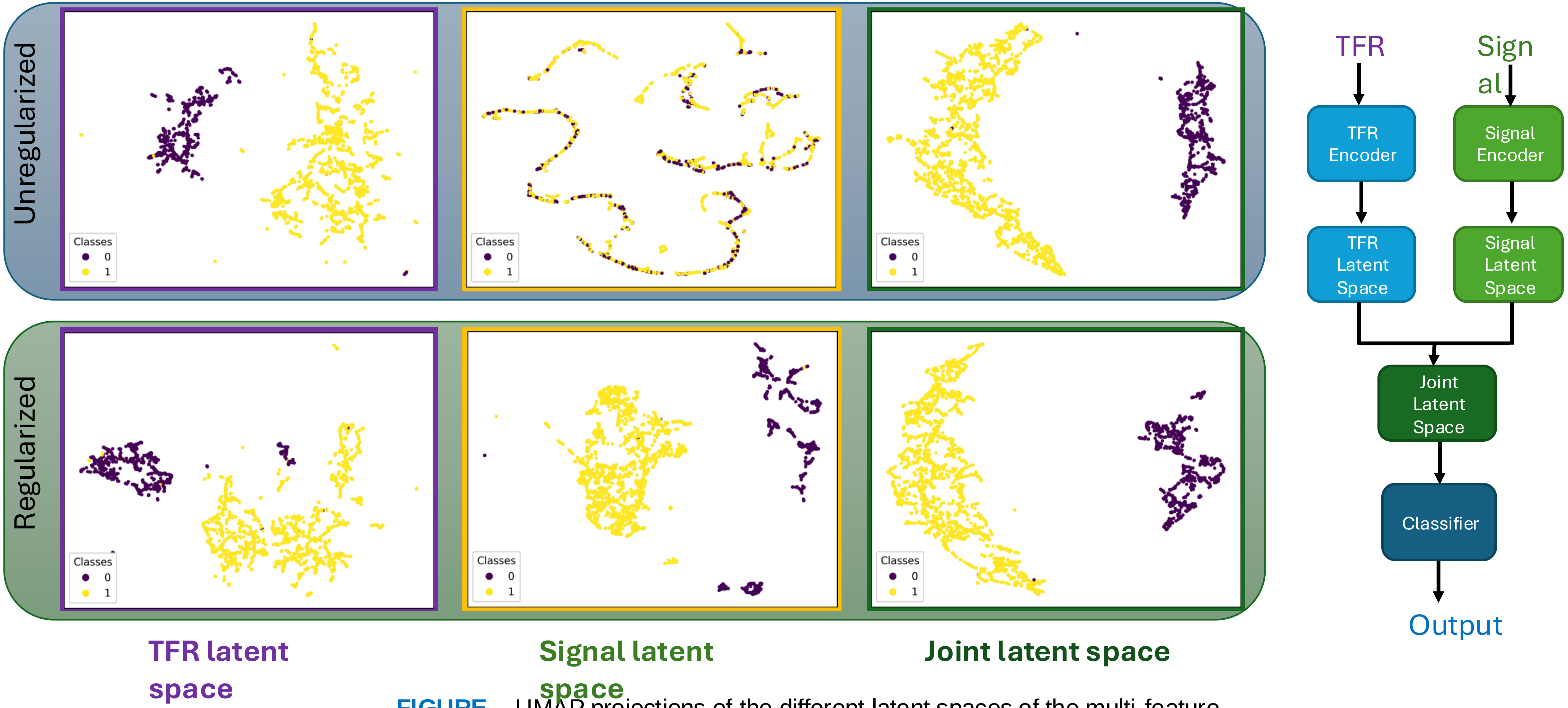
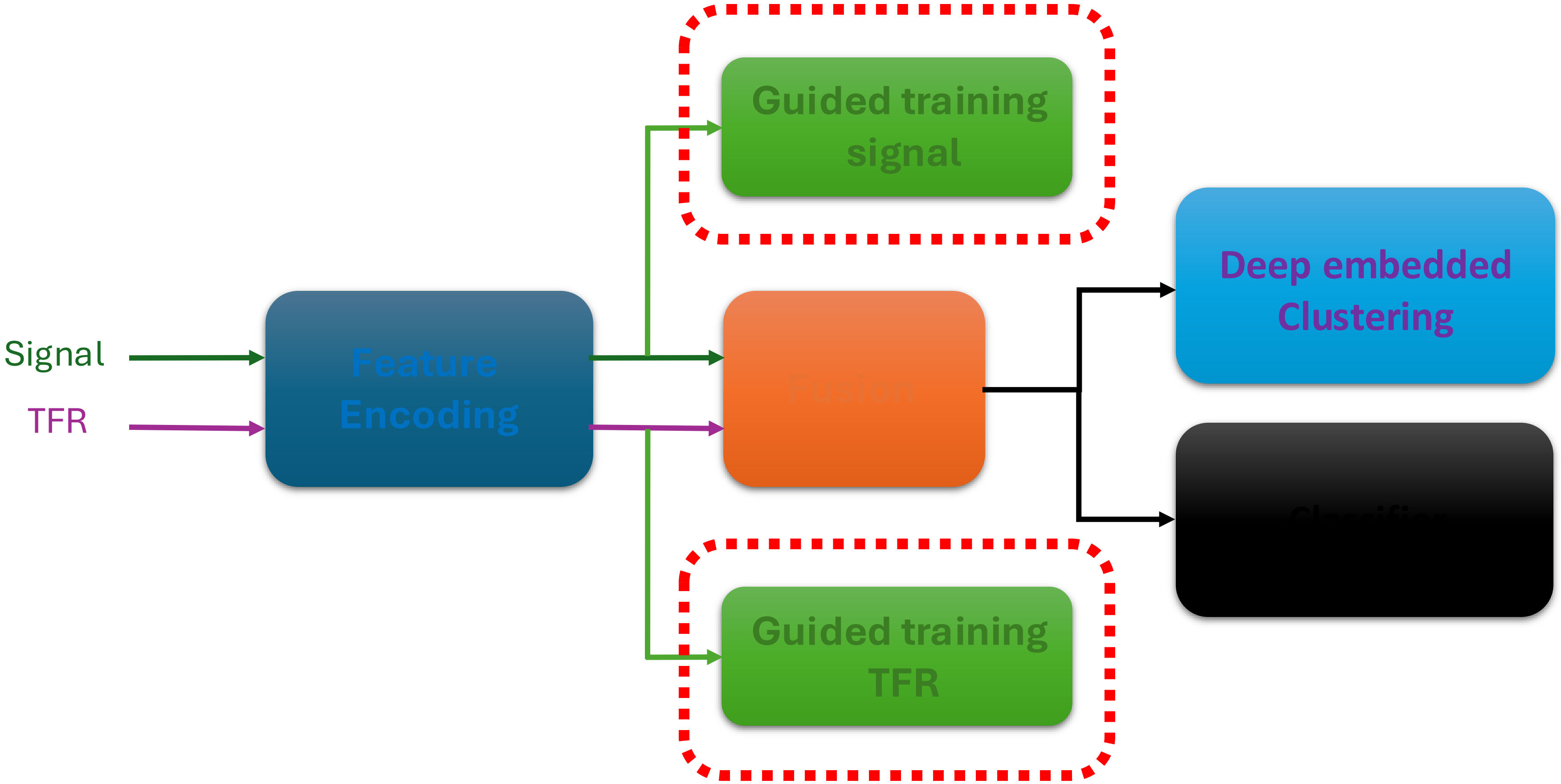


FIGURE - UMAP projections of the different latent spaces of the multi-feature intermediate fusion classification model on the PTB dataset

GDEC


Guided training signal
Signal
TFR
Feature Encoding
Deep embedded Clustering
Guided training TFR

Experiment: influence guided training

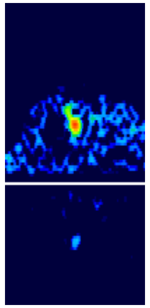
Objective:

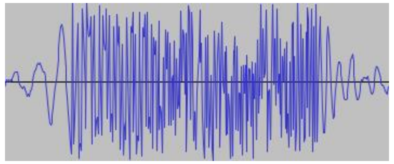
- Influence signal guided training.
- Influence TFR guided training.

Datasets:

HITS:

- TCD Data.
- 1545 samples.
- Three classes.
- Sampling frequency: 4385 Hz.





Metrics:

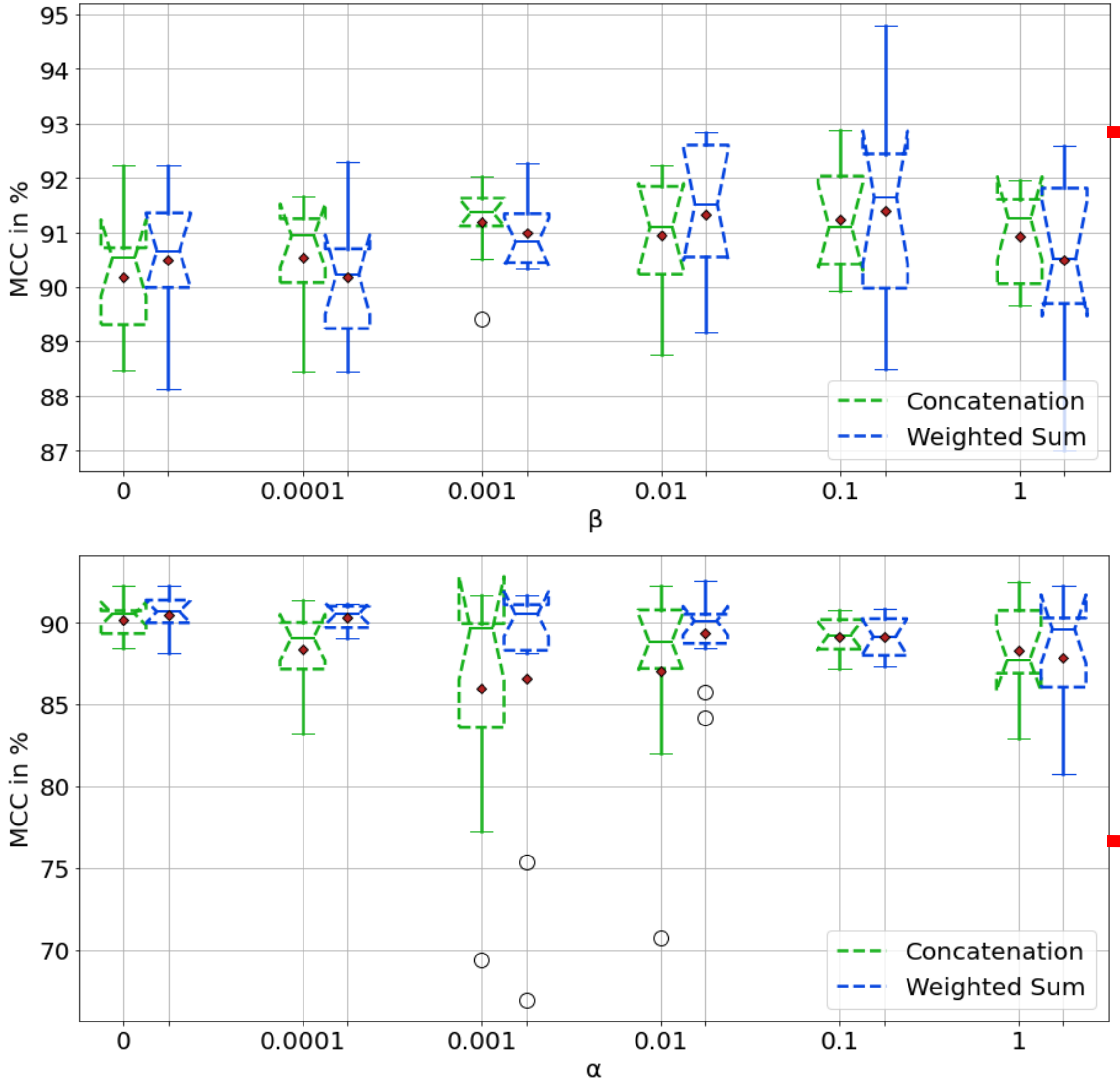
- Mathews Correlation Coefficient (MCC).

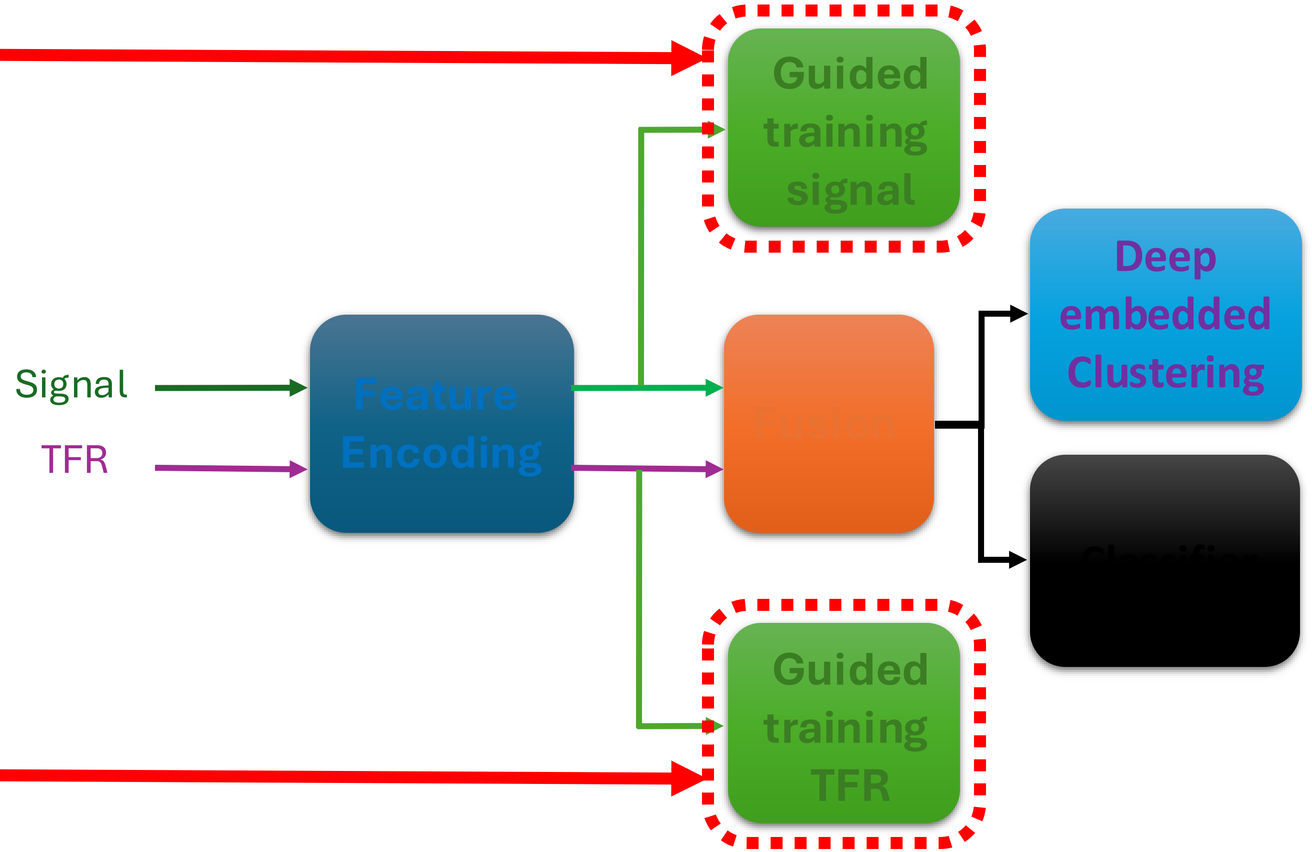
Loss function:

- Cross entropy (CE)

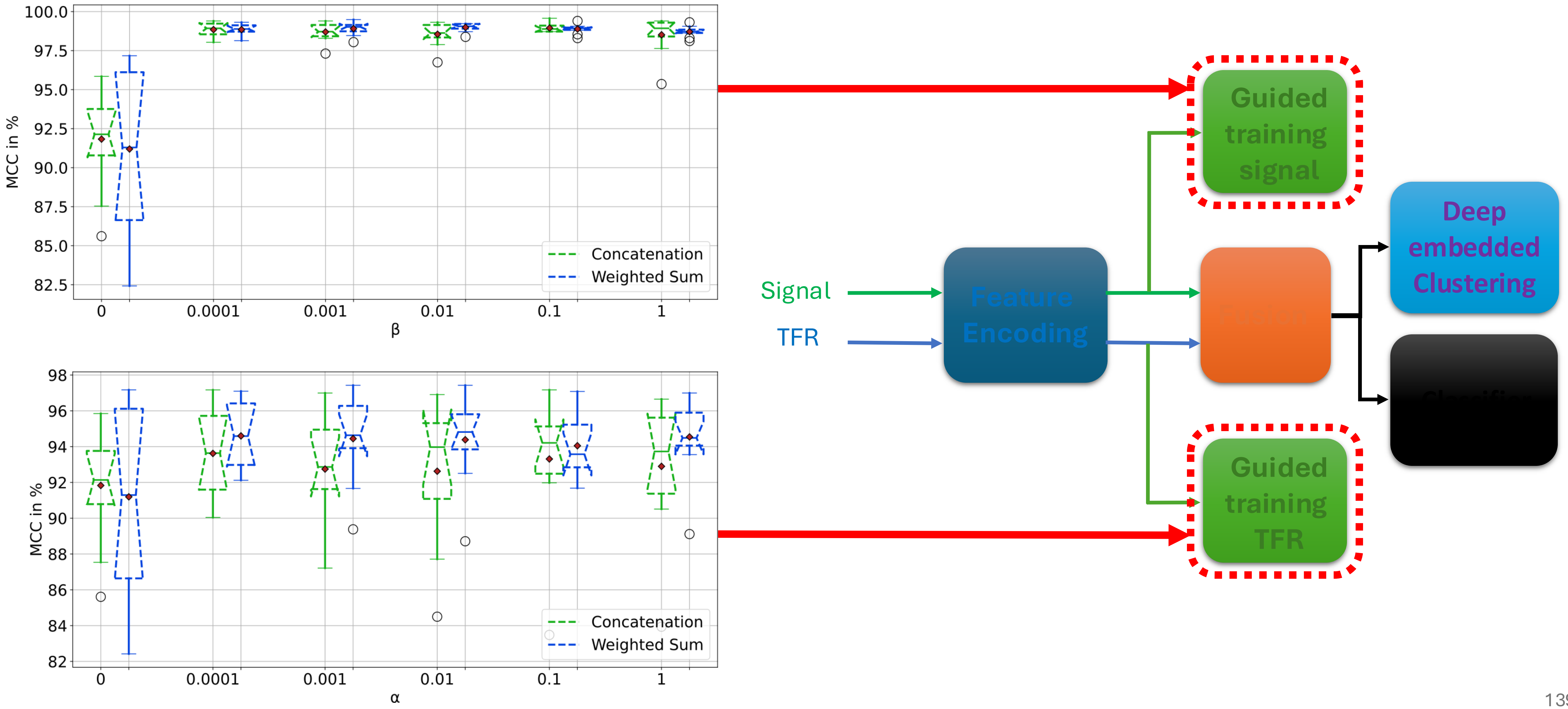
Class	Number of samples
Artifact	403
Gaseous Emboli	569
Solid Emboli	569
Unknown	4

Results: influence guiding HITS Validation

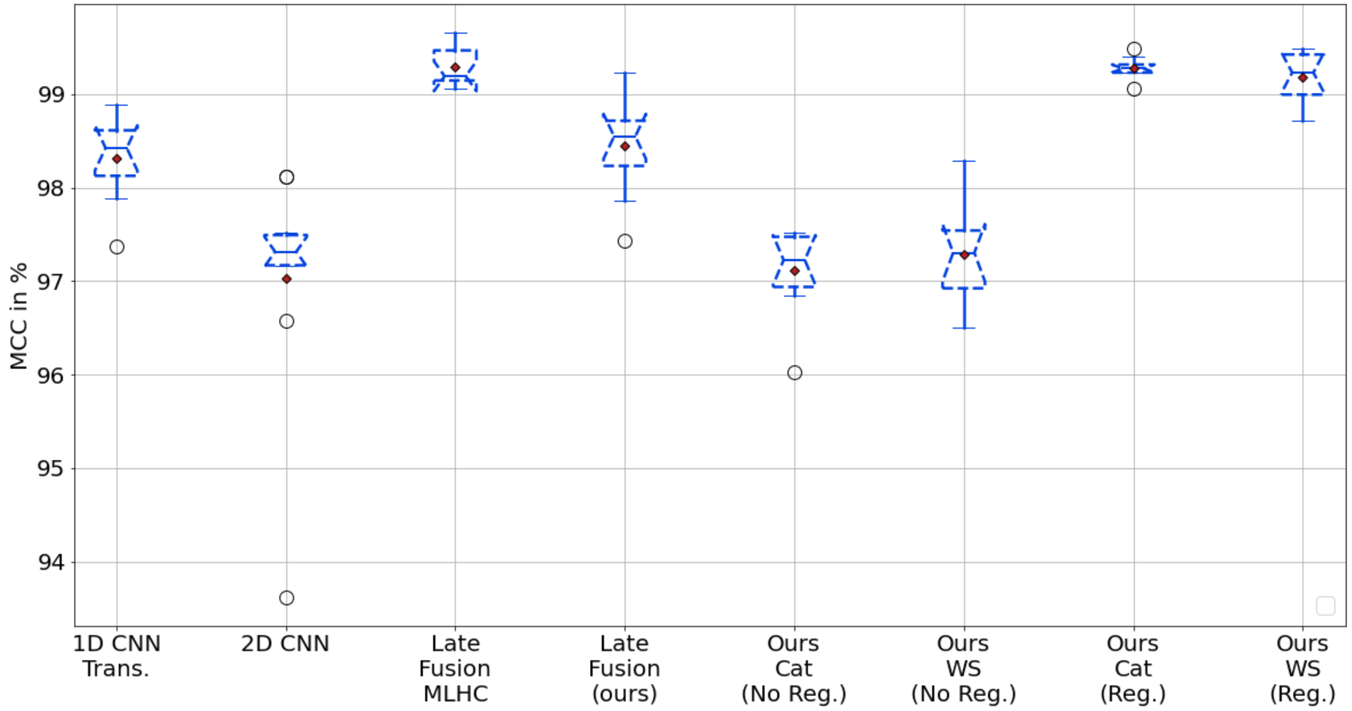




Results: influence guiding PTB



Results Multi-feature classification in PTB



Model	Features	Fusion	F1 Score
1D CNN-Trans.	Raw Signal	-	99.16 ± 0.22
2D CNN	TFR	-	98.51 ± 0.61
Ahmad et al. (2021)	GAF MTF RP		98
Late Fusion (MLHC)		Weight. Sum	99.65 ± 0.10
Late Fusion (ours)			99.22 ± 0.25
Ours (No Reg.)		Cat.	98.60 ± 0.22
Ours (No Reg.)	**Both**	Weight. Sum	98.64 ± 0.25
Ours (Reg.)		**Cat.**	**99.64 ± 0.05**
Ours (Reg.)		Weight. Sum	99.59 ± 0.13

FIGURE - Comparison of the classification performances of different single and multi-feature models on the PTB dataset

Late fusion weights interpretability

Class	Spectrogram	Raw Signal
Artifacts	0.46 ± 0.29	0.54 ± 0.29
Gaseous Emboli	0.65 ± 0.17	0.35 ± 0.17
Solid Emboli	0.71 ± 0.15	0.29 ± 0.15

Attention weights for the HITS dataset

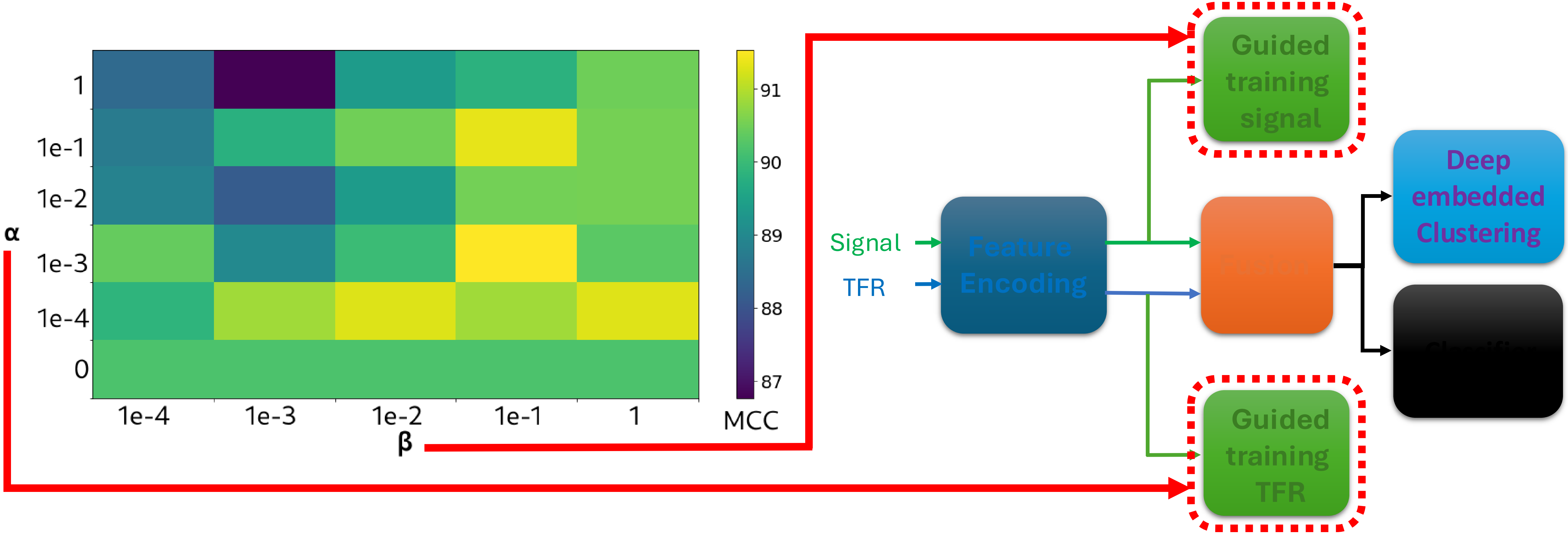
Class	Spectrogram	Raw Signal
Normal	0.49 ± 0.12	0.51 ± 0.12
Abnormal	0.18 ± 0.10	0.82 ± 0.10

Attention weights for the PTB dataset

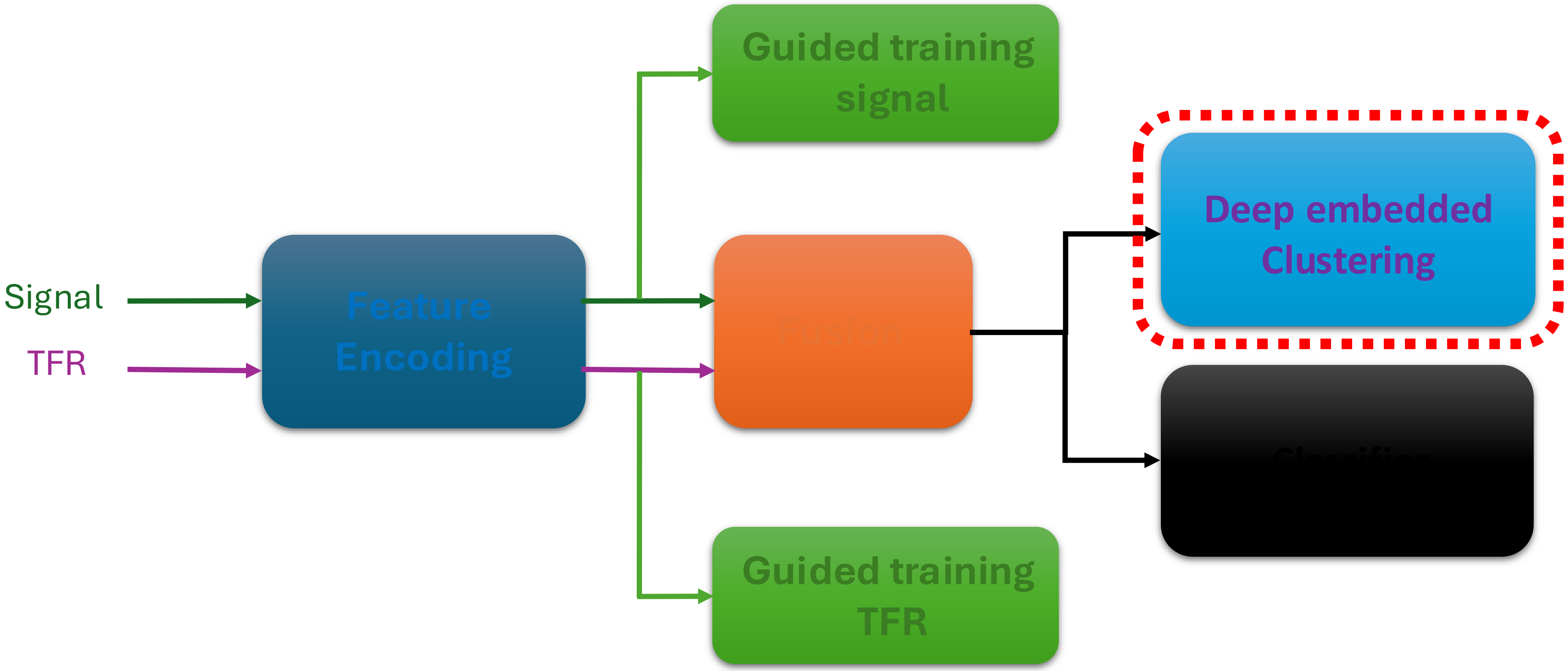
Class	Spectrogram	Raw Signal
N	0.48 ± 0.01	0.52 ± 0.01
S	0.50 ± 0.01	0.50 ± 0.01
V	0.50 ± 0.01	0.50 ± 0.01
F	0.49 ± 0.02	0.51 ± 0.02
Q	0.50 ± 0.003	0.50 ± 0.003

Attention weights for the MIT-BIH dataset

Results: influence guiding HITS both



GDEC


Signal
TFR
Feature Encoding
Guided training signal
Fusion
Guided training TFR
Deep embedded Clustering

Experiment: influence DEC

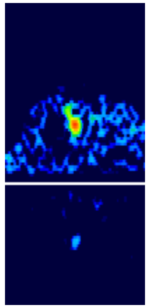
Objective:

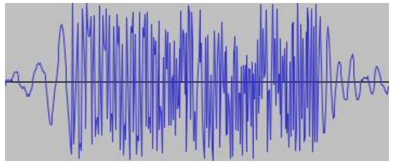
- Influence DEC on the classification performance.

Datasets:

HITS:

- TCD Data.
- 1545 samples.
- Three classes.
- Sampling frequency: 4385 Hz.





Metrics:

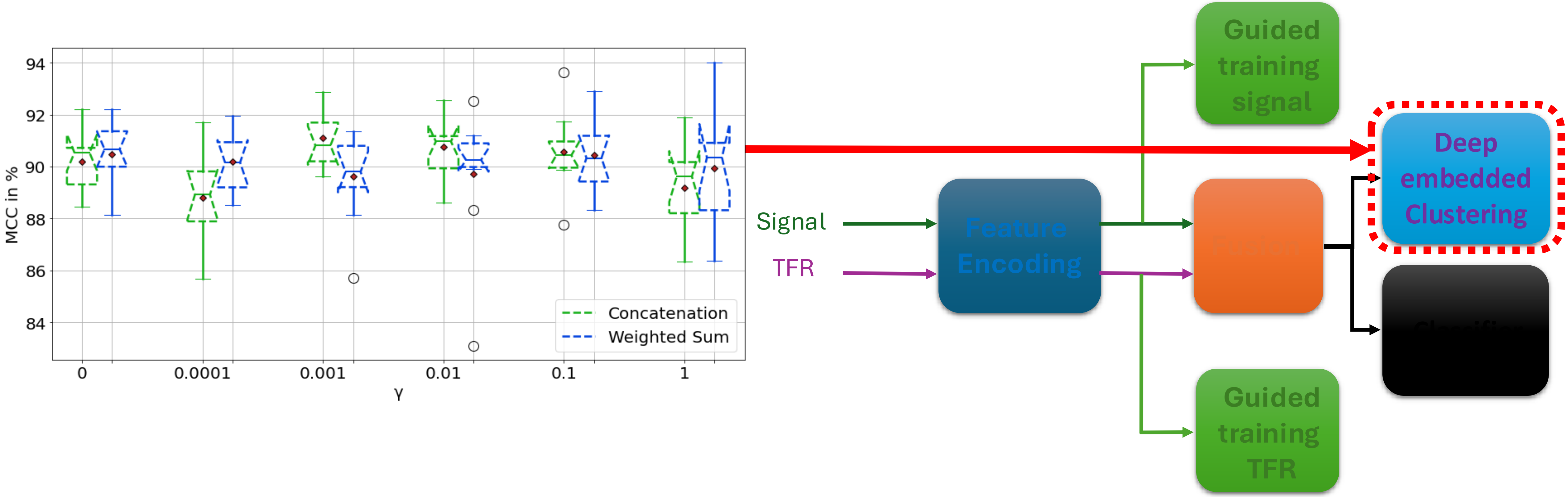
- Mathews Correlation Coefficient (MCC).

Loss function:

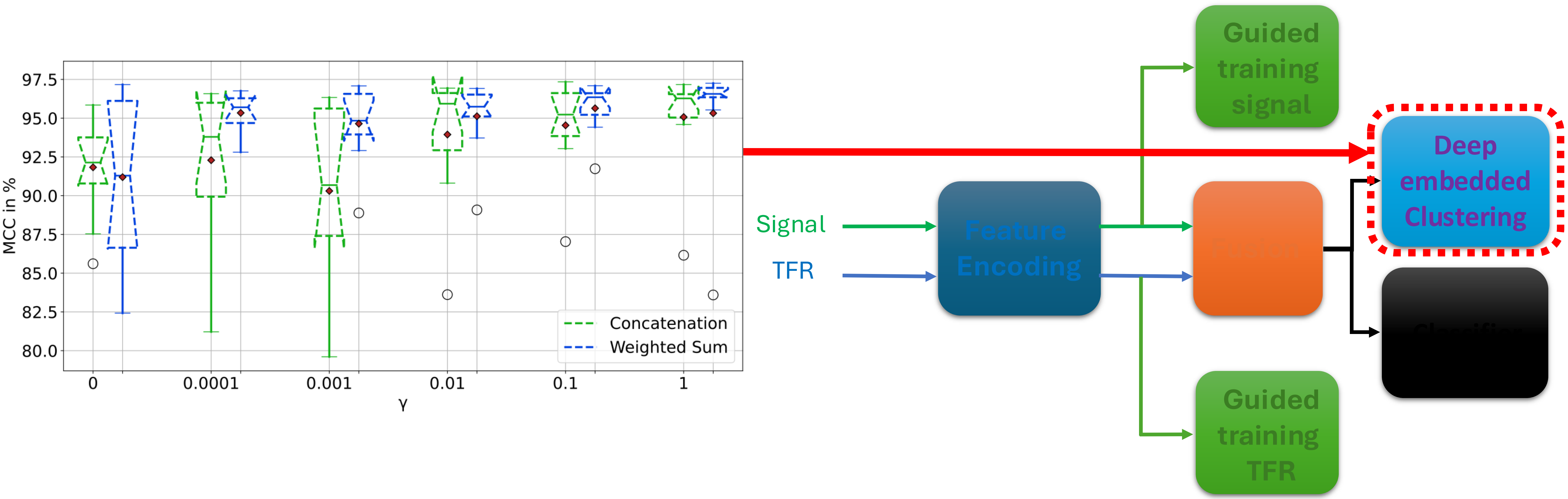
- Cross entropy (CE)

Class	Number of samples
Artifact	403
Gaseous Emboli	569
Solid Emboli	569
Unknown	4

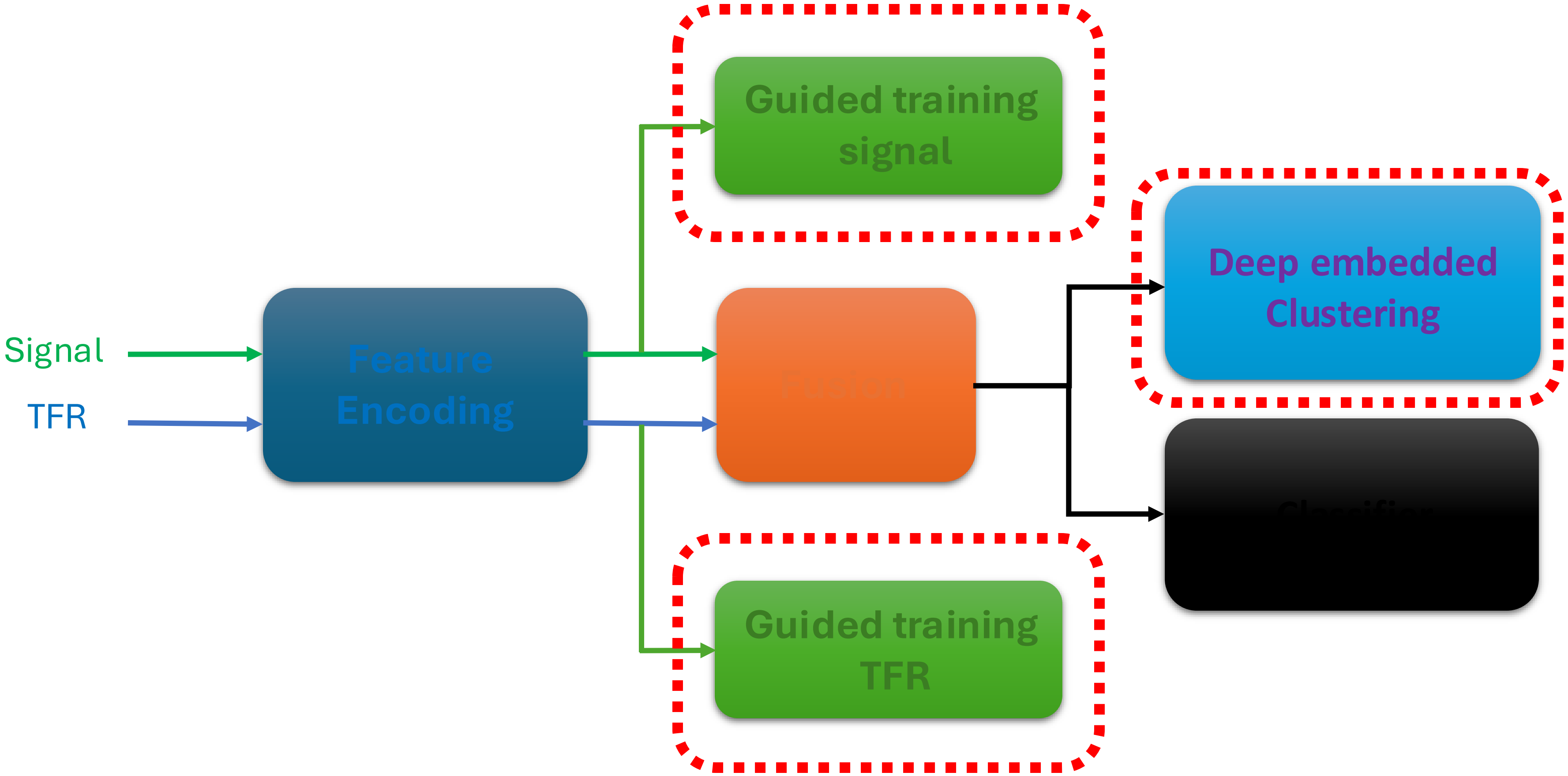
Results: influence DEC HITS validation



Results: influence DEC PTB



GDEC


Signal
TFR
Feature Encoding
Guided training signal
Fusion
Guided training TFR
Deep embedded Clustering

Experiment: noise tolerance on HITS-sada

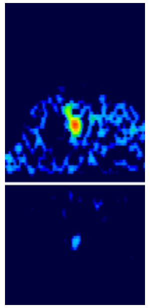
Objective:

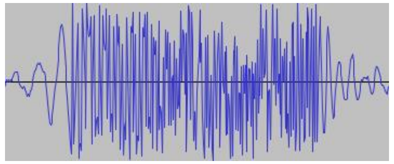
- Study the robustness of GDEC to noisy labeled datasets.

Datasets:

HITS-sada:

- TCD **semi-automatically labeled** data.
- 8 685 samples.
- Three classes.
- Sampling frequency: 4385 Hz.





Metrics:

- Mathews Correlation Coefficient (MCC).
- F1-Score.
- Number of parameters.
- Number of mult-adds.

Loss function:

- **Generalized Cross entropy (GCE).**

Class	Number of samples
Artifact	6 987
Gaseous Emboli	1 002
Solid Emboli	696

Results

Model	MCC	F1-Score	Accuracy	No. Parameters	No. mult-adds (G)
2D CNN	84.03 ± 1.20	86.81 ± 1.50	90.68 ± 1.12	1 681 923	1.23
1D CNN-trans.	85.74 ± 1.16	88.96 ± 0.78	91.35 ± 0.77	**766 271**	**0.173**
MIF-GR	**87.35 ± 0.85**	**89.41 ± 0.64**	**92.59 ± 0.50**	4 833 727	1.40

Table – Multi-feature GDOEs compared to single feature models on a noisy semi-automatically labeled dataset HITS-sada.

Robustness DEC Imbalanced Datasets

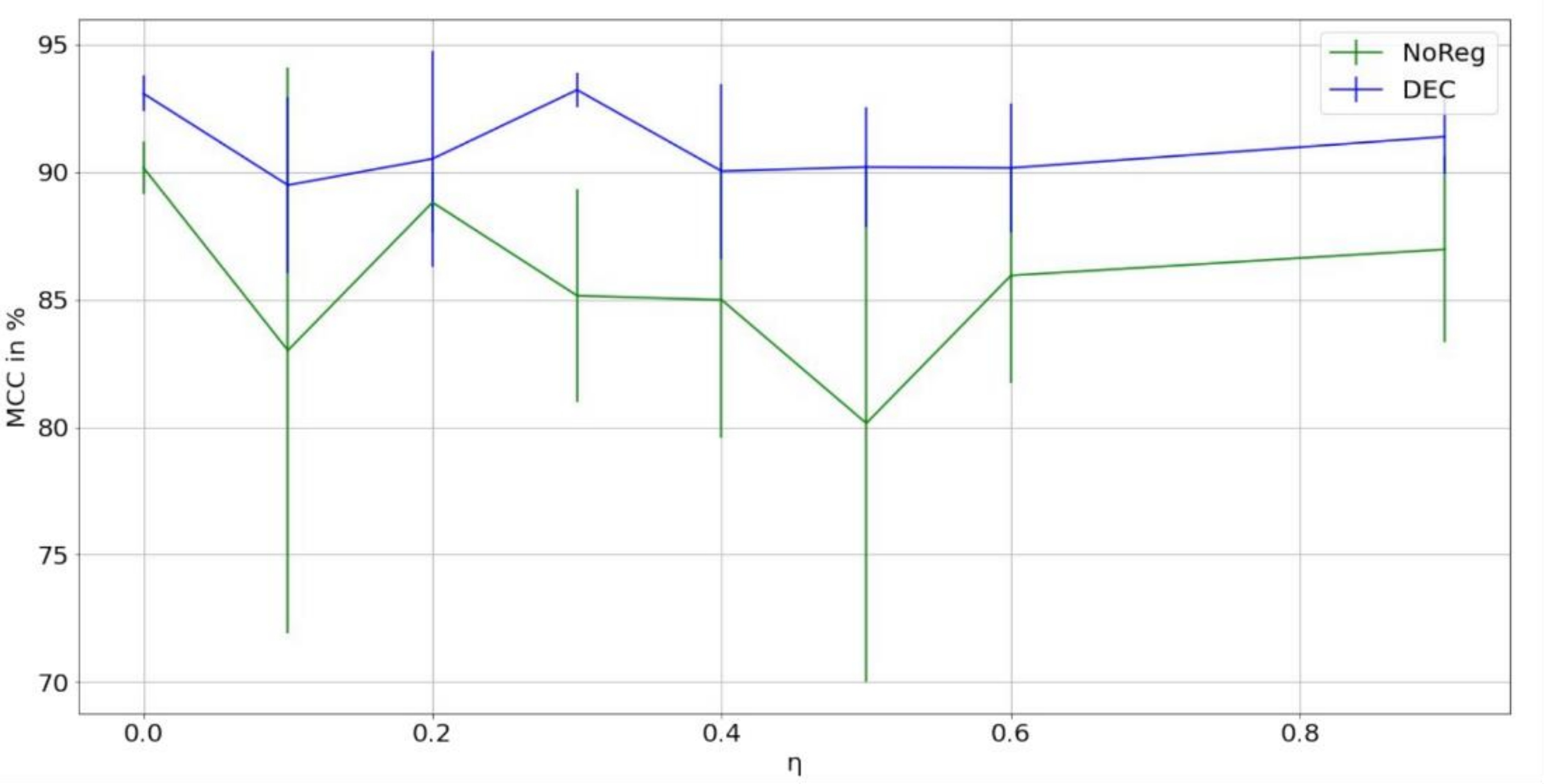


FIGURE - MCC of the multi-feature classification model on the HITS dataset for different levels of label noise.

Robustness DEC Imbalanced Datasets

Noise Rate	Clean Samples	GDEC	Loss	MCC
5	Yes	No	CE	97.08 ± 0.53
		Yes		99.32 ± 0.17
	No	No		88.61 ± 0.74
		Yes		93.13 ± 0.31
	Yes	No	GCE	96.58 ± 0.49
		Yes		98.37 ± 0.27
	No	No		94.10 ± 1.09
		Yes		96.70 ± 0.54

Noise rate of 5 %

Noise Rate	Clean Samples	GDEC	Loss	MCC
10	Yes	No	CE	96.98 ± 0.35
		Yes		99.30 ± 0.19
	No	No		78.55 ± 1.45
		Yes		82.30 ± 1.15
	Yes	No	GCE	96.27 ± 0.59
		Yes		98.57 ± 0.36
	No	No		90.22 ± 1.30
		Yes		91.90 ± 1.18

Noise rate of 10 %

Noise Rate	Clean Samples	GDEC	Loss	MCC
20	Yes	No	CE	96.96 ± 0.54
		Yes		98.99 ± 0.27
	No	No		59.98 ± 1.98
		Yes		63.77 ± 2.08
	Yes	No	GCE	95.66 ± 0.70
		Yes		97.84 ± 0.52
	No	No		71.63 ± 2.52
		Yes		71.66 ± 3.48

Noise rate of 20 %

Table - MCC of the multi-feature classification model on the PTB dataset for different levels of label noise.

Soft labelling

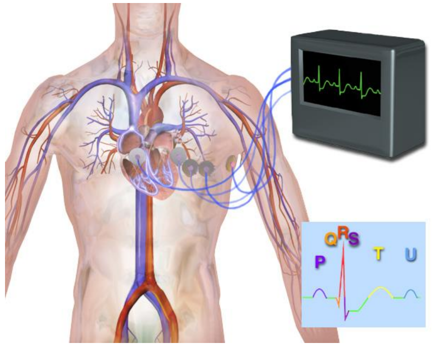
Model	Dataset	Soft Labels	Mean MCC	Median MCC
2D CNN	HITS Small	No	86.13 ± 3.80	87.76 ± 1.85
		Yes	87.03 ± 3.55	87.47 ± 1.75
CNN-Transformer		No	85.92 ± 1.79	86.01 ± 0.82
		Yes	86.52 ± 3.73	87.63 ± 0.86
Hybrid		No	92.50 ± 1.36	92.74 ± 0.95
		Yes	93.12 ± 1.00	92.59 ± 0.11
	HITS Large	No	85.49 ± 0.77	85.49 ± 0.77
		Yes	86.77 ± 0.96	86.35 ± 0.49

Datasets MLHC 2022

HITS:

- TCD Data.
- 1545 samples.
- Three classes.
- Sampling frequency: 4385 Hz.

Class	Number of samples
Artifact	403
Gaseous Emboli	569
Solid Emboli	569



PTB:

- ECG Data.
- 14 552 samples.
- Two classes.
- Sampling frequency: 125 Hz.

Class	Number of samples
Normal	10 506
Abnormal	4 046

MIT-BIH:

- ECG Data.
- 109 436 samples.
- Five classes.
- Sampling frequency: 125 Hz.

Class	Number of samples
N	90 589
S	2 779
V	7 226
F	803

Experiment 1: Advantage of using multiple features
MLHC 2022

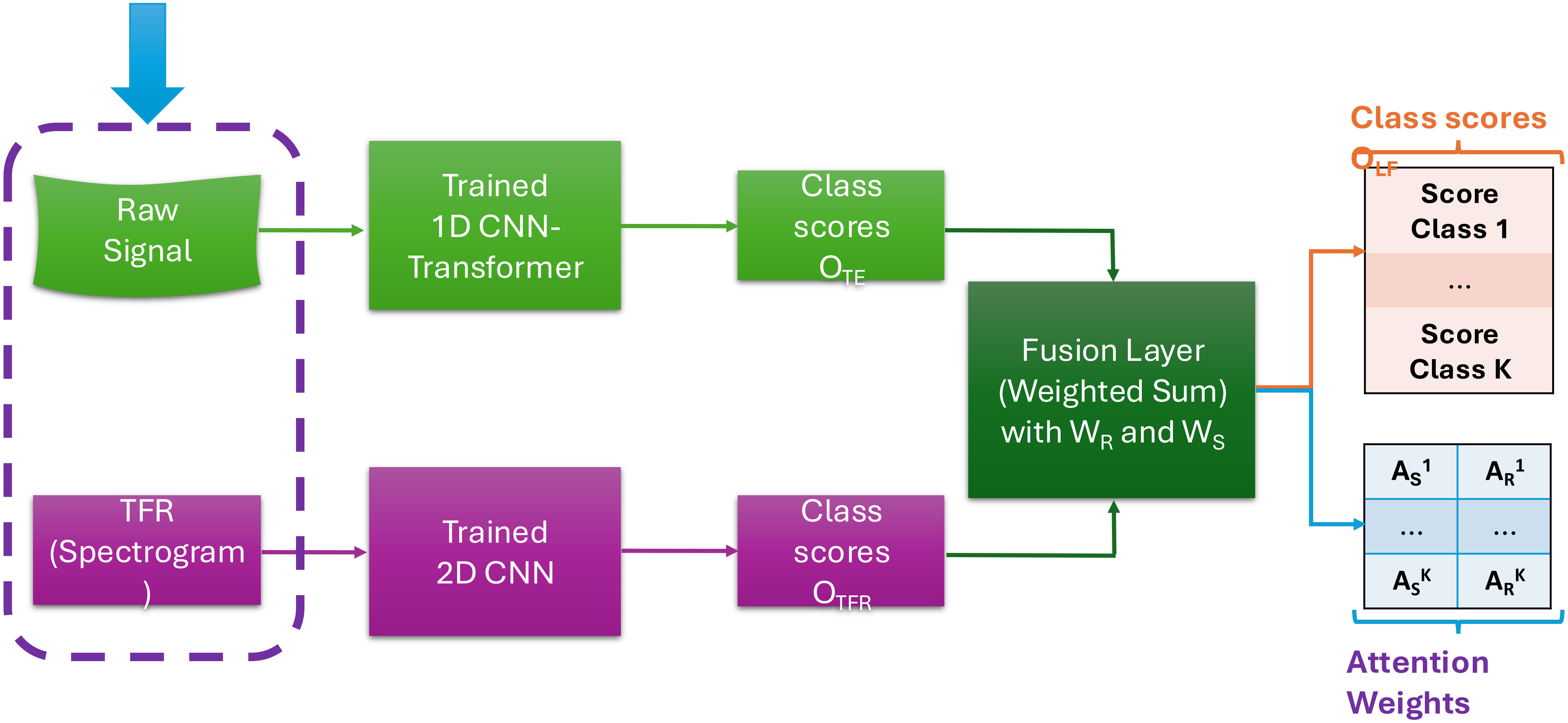


FIGURE - Proposed hybrid CNN Transformer global model

Experimental Setup MLHC 2022

Objective:
- Comparison to **single feature** models.
- Comparison to **SOTA** models.

Models:
- **Single feature models** : 1D CNN-Transformer and 2D CNN.
- **SOTA** : Vindas et al., 2022 (HITS) and Ahmad et al., 2021 (ECG).

Loss function:
- **Cross Entropy** Loss.

Optimizers:
- **ADAM**.
- **NOAM**.

For imbalanced datasets

Metrics:
- Matthews Correlation Coefficient (**MCC**)

Results MLHC 2022

Dataset	Model	MCC	F1-Score	Accuracy
HITS	2D CNN (previous work)	85.53 ± 2.98	85.68 ± 2.31	89.48 ± 2.06
	1DCNN-Transformer	80.29 ± 1.83	85.36 ± 1.09	87.37 ± 1.23
	2D CNN	85.03 ± 3.06	86.88 ± 2.38	90.55 ± 2.12
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	MIF (Ahmad et al., 2021)	-	-	98.4
	MFF (Ahmad et al., 2021)	-	-	99.2
	1DCNN-Transformer	97.92 ± 0.28	98.96 ± 0.14	99.16 ± 0.11
	2D CNN	93.42 ± 2.27	96.66 ± 1.20	97.32 ± 0.91
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	MIF (Ahmad et al., 2021)	-	-	98.6
	MFF (Ahmad et al., 2021)	-	-	**99.7**
	1DCNN-Transformer	93.17 ± 0.70	89.44 ± 0.99	97.87 ± 0.24

Results MLHC 2022

Dataset	Model	MCC	F1-Score	Accuracy
HITS	2D CNN (previous work)	85.53 ± 2.98	85.68 ± 2.31	89.48 ± 2.06
	1DCNN-Transformer	80.29 ± 1.83	85.36 ± 1.09	87.37 ± 1.23
	2D CNN	85.03 ± 3.06	86.88 ± 2.38	90.55 ± 2.12
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	MIF (Ahmad et al., 2021)	-	-	98.4
	MFF (Ahmad et al., 2021)	-	-	99.2
	1DCNN-Transformer	97.92 ± 0.28	98.96 ± 0.14	99.16 ± 0.11
	2D CNN	93.42 ± 2.27	96.66 ± 1.20	97.32 ± 0.91
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	MIF (Ahmad et al., 2021)	-	-	98.6
	MFF (Ahmad et al., 2021)	-	-	99.7
	1DCNN-Transformer	93.17 ± 0.70	89.44 ± 0.99	97.87 ± 0.24

Using both representations increase the classification performances of the model in the three datasets

Results MLHC 2022

Dataset	Model	MCC	F1-Score	Accuracy
HITS	2D CNN (previous work)	85.53 ± 2.98	85.68 ± 2.31	89.48 ± 2.06
	1DCNN-Transformer	80.29 ± 1.83	85.36 ± 1.09	87.37 ± 1.23
	2D CNN	85.03 ± 3.06	86.88 ± 2.38	90.55 ± 2.12
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	MIF (Ahmad et al., 2021)	-	-	98.4
	MFF (Ahmad et al., 2021)	-	-	99.2
	1DCNN-Transformer	97.92 ± 0.28	98.96 ± 0.14	99.16 ± 0.11
	2D CNN	93.42 ± 2.27	96.66 ± 1.20	97.32 ± 0.91
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	MIF (Ahmad et al., 2021)	-	-	98.6
	MFF (Ahmad et al., 2021)	-	-	**99.7**
	1DCNN-Transformer	93.17 ± 0.70	89.44 ± 0.99	97.87 ± 0.24

More stable models (reduced variability), except for the HITS dataset.

Results MLHC 2022

Dataset	Model	MCC	F1-Score	Accuracy
HITS	2D CNN (previous work)	85.53 ± 2.98	85.68 ± 2.31	89.48 ± 2.06
	1DCNN-Transformer	80.29 ± 1.83	85.36 ± 1.09	87.37 ± 1.23
	2D CNN	85.03 ± 3.06	86.88 ± 2.38	90.55 ± 2.12
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	MIF (Ahmad et al., 2021)	-	-	98.4
	MFF (Ahmad et al., 2021)	-	-	99.2
	1DCNN-Transformer	97.92 ± 0.28	98.96 ± 0.14	99.16 ± 0.11
	2D CNN	93.42 ± 2.27	96.66 ± 1.20	97.32 ± 0.91
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	MIF (Ahmad et al., 2021)	-	-	98.6
	MFF (Ahmad et al., 2021)	-	-	**99.7**
	1DCNN-Transformer	93.17 ± 0.70	89.44 ± 0.99	97.87 ± 0.24

State-of-the-art results on two datasets.

Experiment 2: Influence of the fusion layer
MLHC 2022

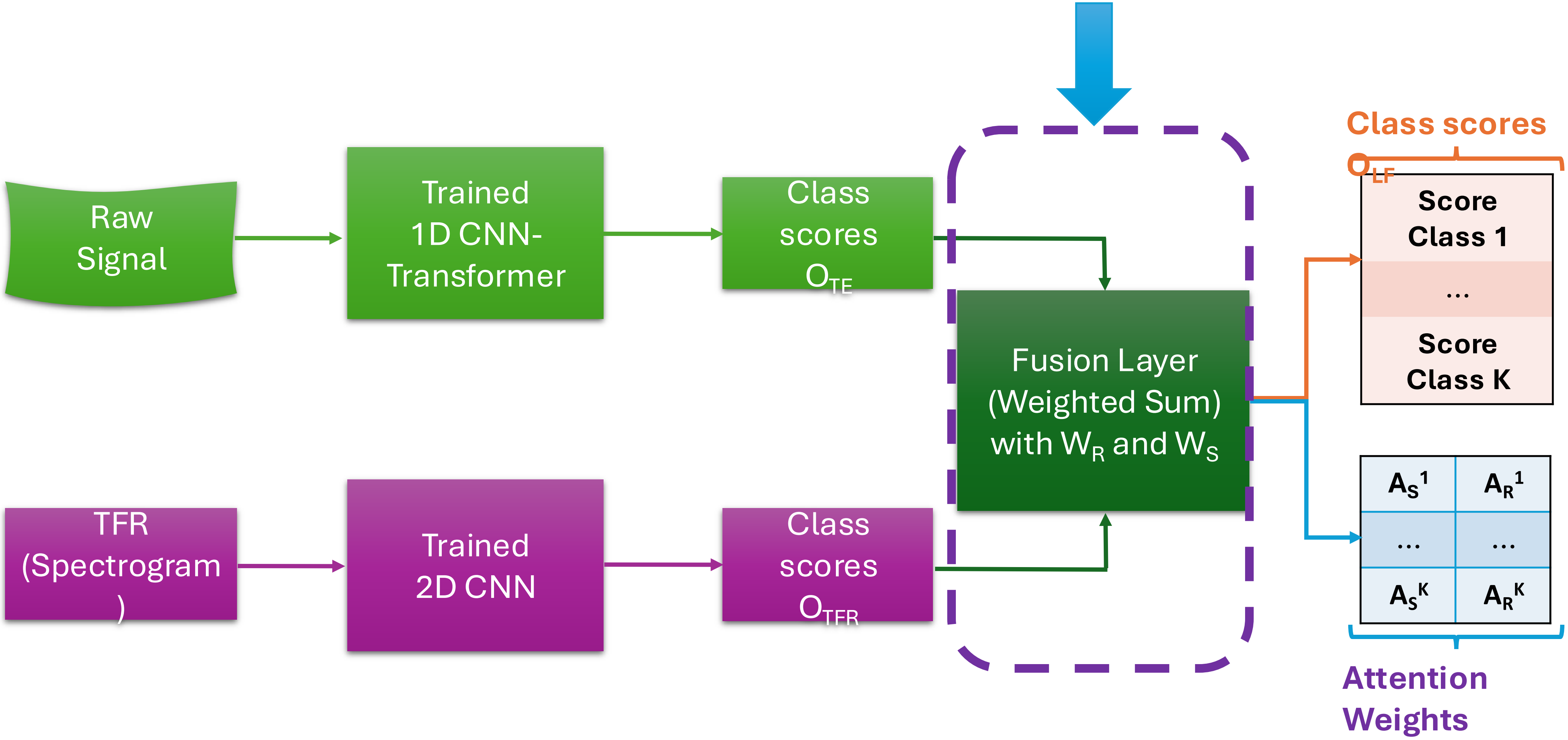


FIGURE - Proposed hybrid CNN Transformer global model

Experimental Setup MLHC 2022

Objective:

- Comparison to **intermediate fusion** models.

Models:

- **Concatenation**.
- **Sum**.
- **Weighted sum**.

Loss function:

- **Cross Entropy** Loss.

Optimizers:

- **NOAM**.

For imbalanced datasets

Metrics:

- Matthews Correlation Coefficient (**MCC**).
- **F1-Score**.
- **Accuracy**.

Results MLHC 2022

Dataset	Fusion Type	MCC	F1-Score	Accuracy
HITS	Concatenation	84.96 ± 2.54	86.37 ± 2.11	90.62 ± 1.65
	Sum	89.04 ± 1.98	90.23 ± 1.71	93.16 ± 1.29
	Weighted Sum	86.31 ± 2.80	87.73 ± 2.32	91.31 ± 1.92
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	Concatenation	92.91 ± 2.61	96.42 ± 1.33	97.11 ± 1.05
	Sum	92.12 ± 2.33	96.02 ± 1.19	96.78 ± 0.99
	Weighted Sum	92.74 ± 2.01	96.35 ± 1.00	97.06 ± 0.81
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	Concatenation	91.51 ± 0.79	86.93 ± 1.10	97.42 ± 0.27
	Sum	91.89 ± 0.47	87.50 ± 0.87	97.55 ± 0.15
	Weighted Sum	91.56 ± 0.72	86.70 ± 1.13	97.44 ± 0.24
	Hybrid	**94.63 ± 0.29**	**91.28 ± 0.54**	98.37 ± 0.09

Results MLHC 2022

Dataset	Fusion Type	MCC	F1-Score	Accuracy
HITS	Concatenation	84.96 ± 2.54	86.37 ± 2.11	90.62 ± 1.65
	Sum	89.04 ± 1.98	90.23 ± 1.71	93.16 ± 1.29
	Weighted Sum	86.31 ± 2.80	87.73 ± 2.32	91.31 ± 1.92
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	Concatenation	92.91 ± 2.61	96.42 ± 1.33	97.11 ± 1.05
	Sum	92.12 ± 2.33	96.02 ± 1.19	96.78 ± 0.99
	Weighted Sum	92.74 ± 2.01	96.35 ± 1.00	97.06 ± 0.81
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	Concatenation	91.51 ± 0.79	86.93 ± 1.10	97.42 ± 0.27
	Sum	91.89 ± 0.47	87.50 ± 0.87	97.55 ± 0.15
	Weighted Sum	91.56 ± 0.72	86.70 ± 1.13	97.44 ± 0.24
The best fus	**Hybrid**	**94.63 ± 0.29**	**91.28 ± 0.54**	98.37 ± 0.09

Results MLHC 2022

Dataset	Fusion Type	MCC	F1-Score	Accuracy
HITS	Concatenation	84.96 ± 2.54	86.37 ± 2.11	90.62 ± 1.65
	Sum	89.04 ± 1.98	90.23 ± 1.71	93.16 ± 1.29
	Weighted Sum	86.31 ± 2.80	87.73 ± 2.32	91.31 ± 1.92
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	Concatenation	92.91 ± 2.61	96.42 ± 1.33	97.11 ± 1.05
	Sum	92.12 ± 2.33	96.02 ± 1.19	96.78 ± 0.99
	Weighted Sum	92.74 ± 2.01	96.35 ± 1.00	97.06 ± 0.81
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	Concatenation	91.51 ± 0.79	86.93 ± 1.10	97.42 ± 0.27
	Sum	91.89 ± 0.47	87.50 ± 0.87	97.55 ± 0.15
	Weighted Sum	91.56 ± 0.72	86.70 ± 1.13	97.44 ± 0.24
	Hybrid	**94.63 ± 0.29**	**91.28 ± 0.54**	98.37 ± 0.09

For the HITS dataset: the intermediate sum fusion method achieve similar performances as the late hybrid fusion method

Results MLHC 2022

Dataset	Fusion Type	MCC	F1-Score	Accuracy
HITS	Concatenation	84.96 ± 2.54	86.37 ± 2.11	90.62 ± 1.65
	Sum	89.04 ± 1.98	90.23 ± 1.71	93.16 ± 1.29
	Weighted Sum	86.31 ± 2.80	87.73 ± 2.32	91.31 ± 1.92
	Hybrid	**89.33 ± 2.77**	**91.15 ± 1.97**	**93.39 ± 1.74**
PTB	Concatenation	92.91 ± 2.61	96.42 ± 1.33	97.11 ± 1.05
	Sum	92.12 ± 2.33	96.02 ± 1.19	96.78 ± 0.99
	Weighted Sum	92.74 ± 2.01	96.35 ± 1.00	97.06 ± 0.81
	Hybrid	**99.29 ± 0.21**	**99.65 ± 0.10**	**99.71 ± 0.08**
MIT-BIH	Concatenation	91.51 ± 0.79	86.93 ± 1.10	97.42 ± 0.27
	Sum	91.89 ± 0.47	87.50 ± 0.87	97.55 ± 0.15
	Weighted Sum	91.56 ± 0.72	86.70 ± 1.13	97.44 ± 0.24
	Hybrid	**94.63 ± 0.29**	**91.28 ± 0.54**	98.37 ± 0.09

- The other fusion methods have similar performances for the three datasets.
- Worst performances than the best single feature model of experiment 1.

Results MLHC 2022

Class	Spectrogram	Raw Signal
Artifacts	0.46 ± 0.29	0.54 ± 0.29
Gaseous Emboli	0.65 ± 0.17	0.35 ± 0.17
Solid Emboli	0.71 ± 0.15	0.29 ± 0.15

Attention weights for the HITS dataset

Class	Spectrogram	Raw Signal
Normal	0.49 ± 0.12	0.51 ± 0.12
Abnormal	0.18 ± 0.10	0.82 ± 0.10

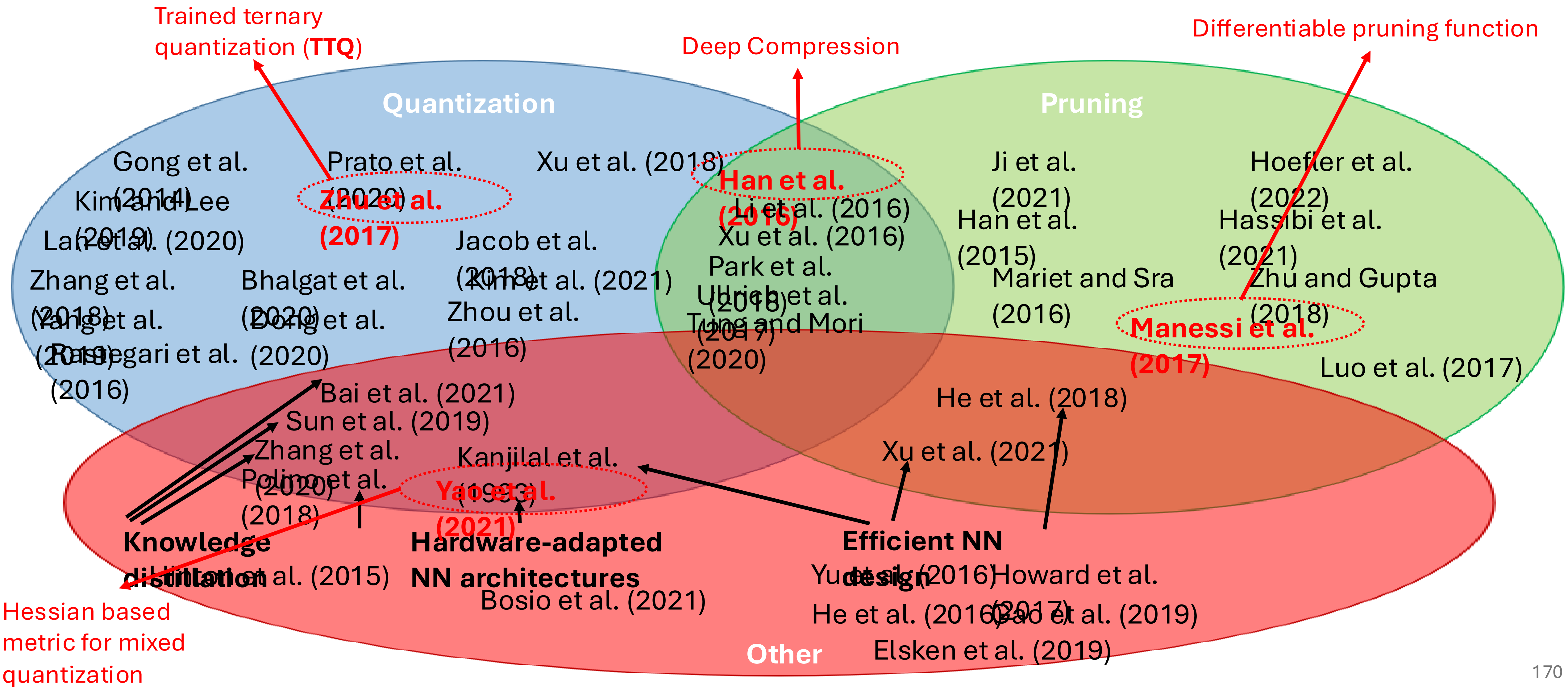
Attention weights for the PTB dataset

Class	Spectrogram	Raw Signal
N	0.48 ± 0.01	0.52 ± 0.01
S	0.50 ± 0.01	0.50 ± 0.01
V	0.50 ± 0.01	0.50 ± 0.01
F	0.49 ± 0.02	0.51 ± 0.02
Q	0.50 ± 0.003	0.50 ± 0.003

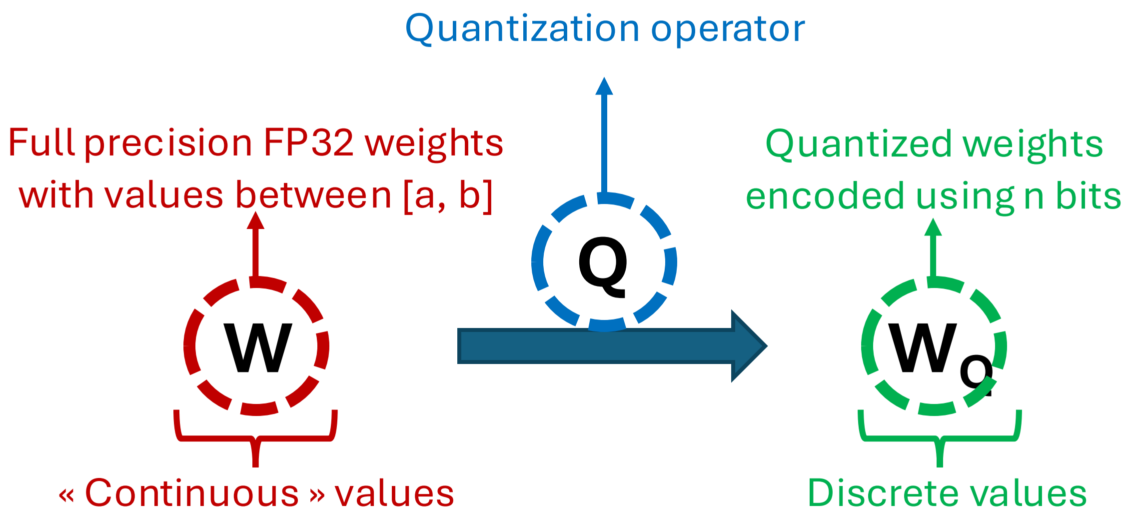
Attention weights for the MIT-BIH dataset

Contribution 3 : Model compression based on extreme quantization

General Overview

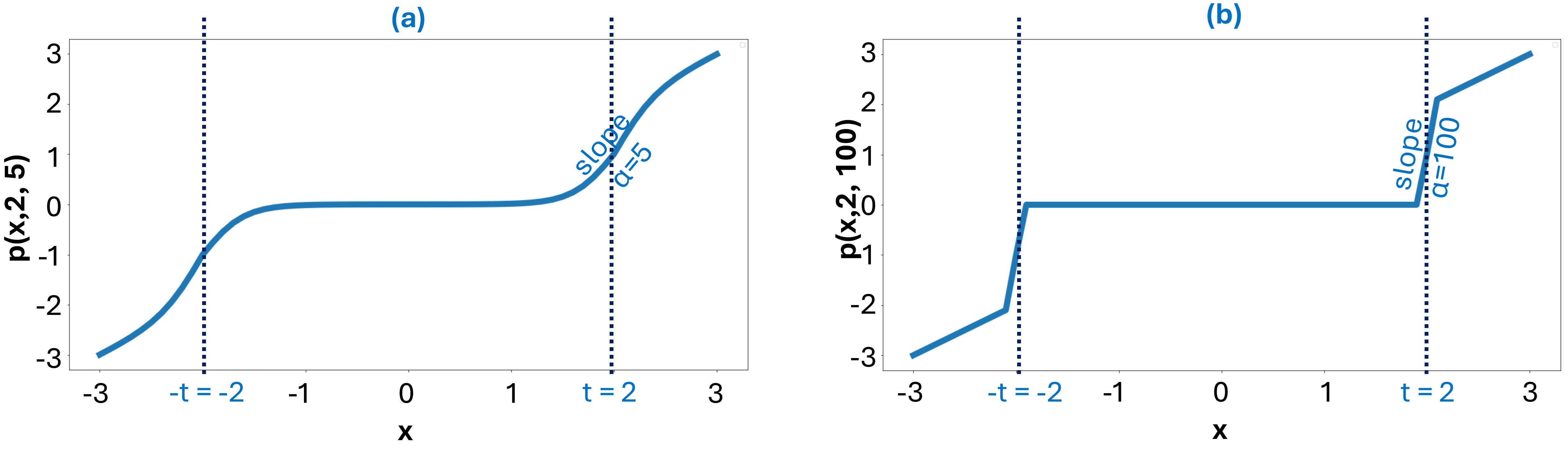


Quantization principle



- **Clipping range** : interval [a, b] where the values of **W** live.
- **Calibration** : step of clipping range search.
- **Scaling factor S** : Number of partitions of the clipping range to use.

Differentiable pruning function



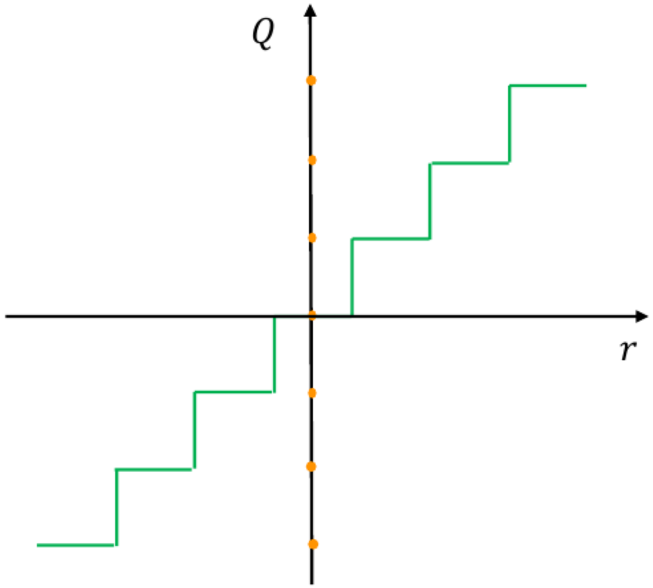
$$\forall x, t, \alpha \in \mathbb{R}, p(x; t, \alpha) = [ReLU(x - t) + t \times \sigma(\alpha \times (x - t)] + [-ReLU(-x - t) - t \times \sigma(\alpha \times (-x - t))]$$

Figure – Differentiable pruning function (**Manessi et al. 2017**)

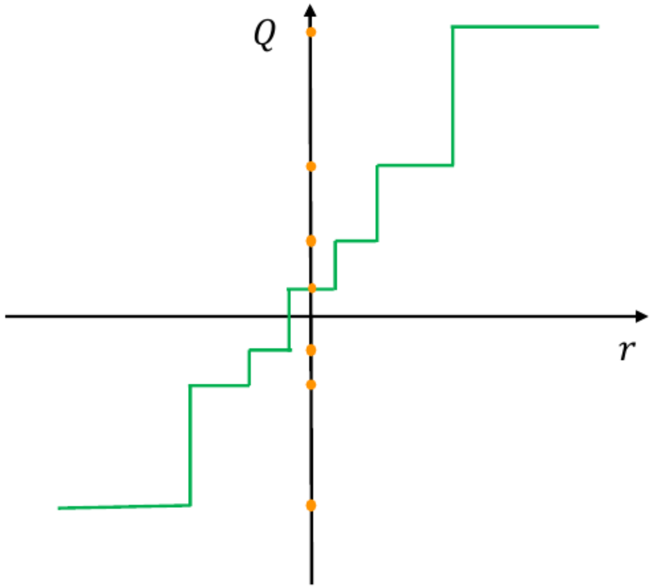
Quantization

Uniform vs non-uniform :

+ Easier to deploy
- Worst classification performances



Uniform

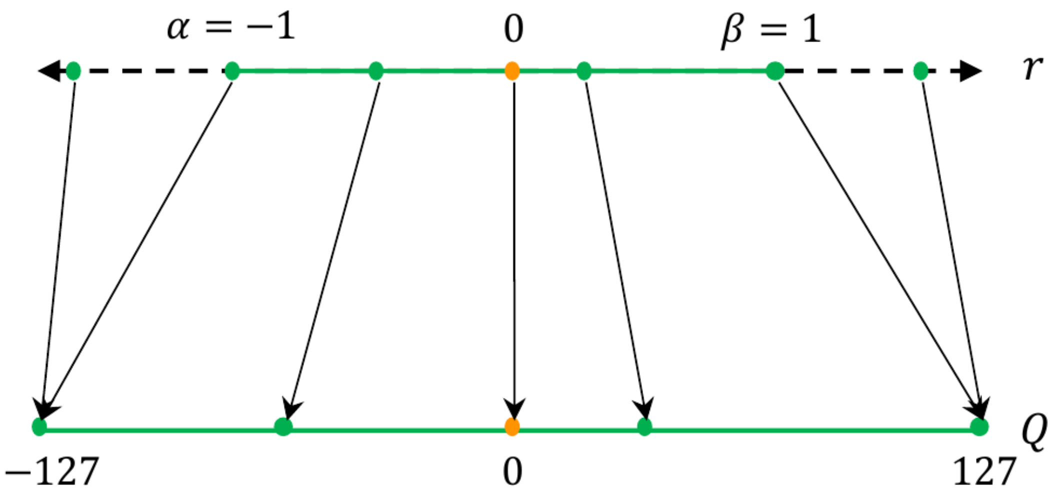


Non-uniform

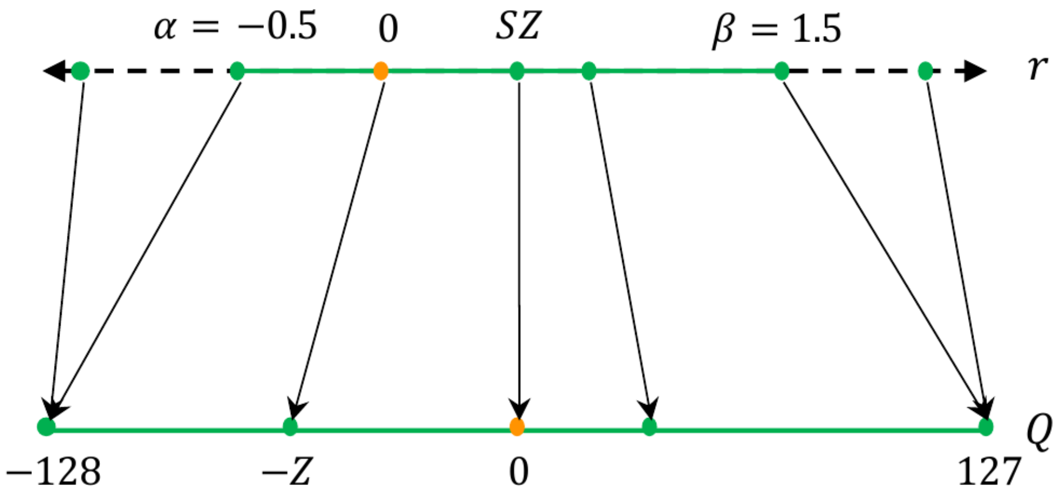
+ Higher classification performances
- More difficult to deploy

Symmetric vs asymmetric :

Symmetri



Asymmetric



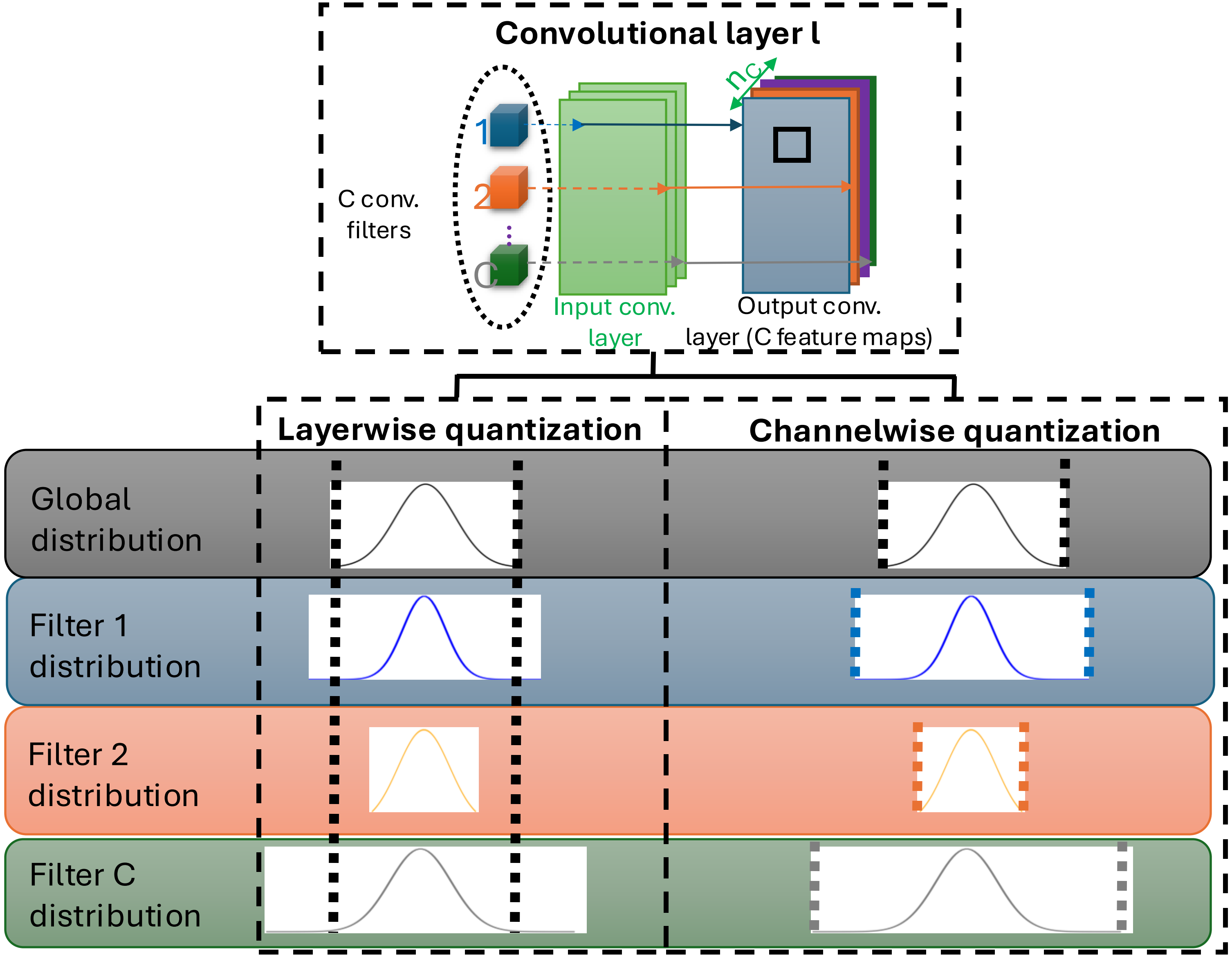
+ Easier to implement
+ Reduce computational cost
- Not adapted to imbalanced weights/activations
- Worst classification performances

+ Better classification performances
+ Adapted for imbalanced weights/activations
- More difficult to implement
- More computationally expensive

Static vs dynamic (w.r.t. clipping range) :

Static	Dynamic
Clipping range pre-computed before inference	Clipping range computed dynamically during inference
+ Less computation resources	+ Higher performances
- Lower performances	- Computationally expensive
==> Most commonly used	

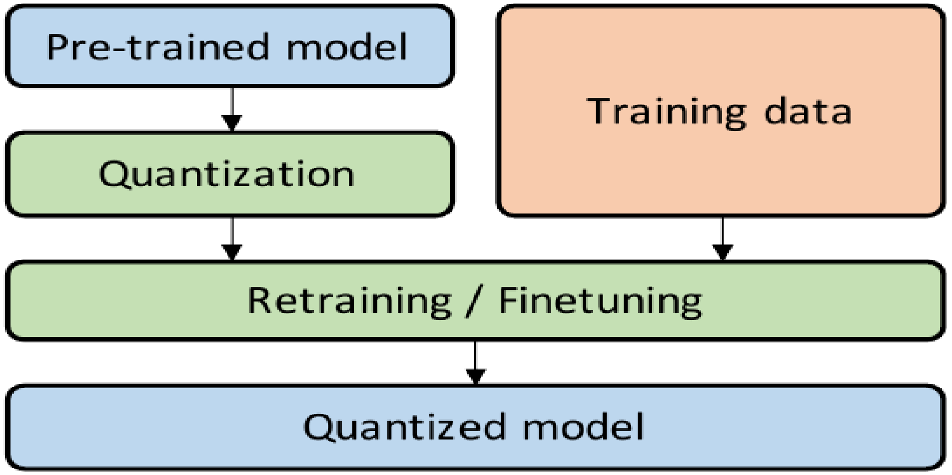
Quantization granularities



Quantization

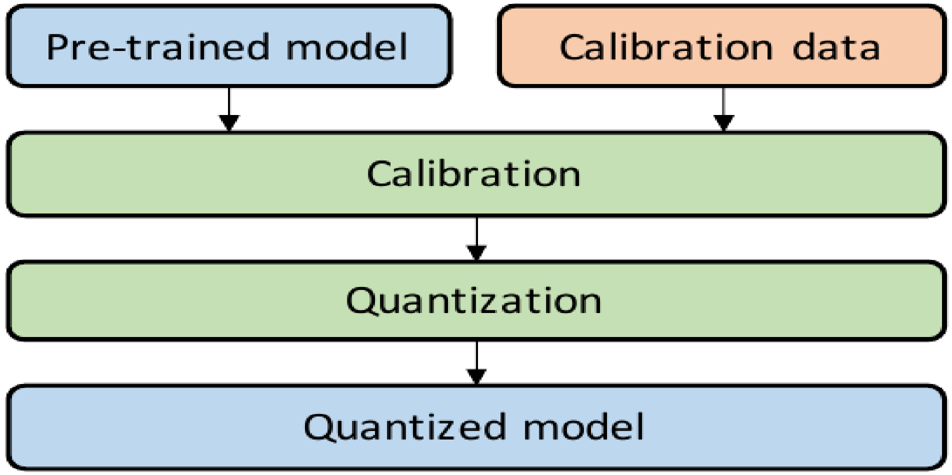
QAT vs PTQ vs ZSQ :

Quantized Aware Training



+ Higher classification performances
~ Be careful with gradient computation
- Expensive during training

Post-Training Quantization



+ Does not modify the training procedure
- Lower classification performances

Types of quantization approaches (Gholami et al. 2021)

- QAT vs PTQ vs ZSQ :
  - Zero-Shot Quantization:
    - No need of training, validation or testing data.
    - Good when we do not have access to the original training data.
  - Can be mixed with QAT and PTQ:
    - No data + fine-tuning ==> ZSQ + PTQ.
      - Correcting biases introduced in the quantized weights.
    - No data + fine-tuning ==> ZSQ + QAT
      - Ex.: use of synthetic data.

Stochastic vs deterministic :

Stochastic

$$Q(W) = \begin{cases} W_Q \text{ with probability } p \\ W_{Q'} \text{ with probability } 1-p \end{cases}$$

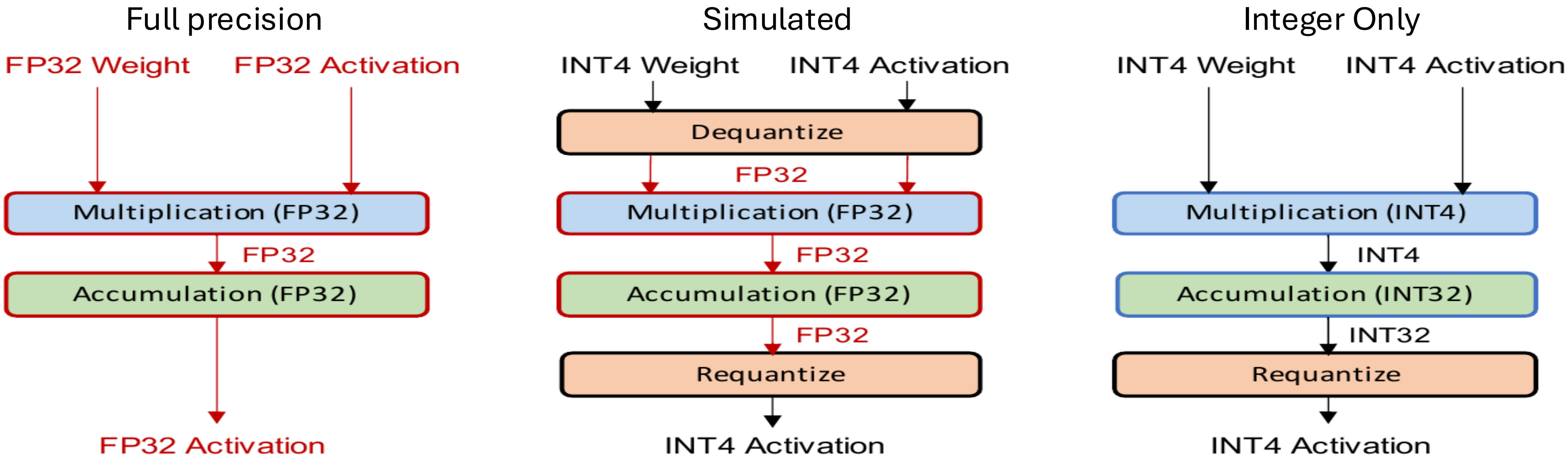
+ Higher classification performances
~ Choice of the stochastic strategy
- Overhead due to the generation of random numbers

Deterministic

$$Q(W) = W_Q$$

+ Less computationally expensive
+ “Easier” to optmize
- Lower classification performances

Simulated vs Integer-only :



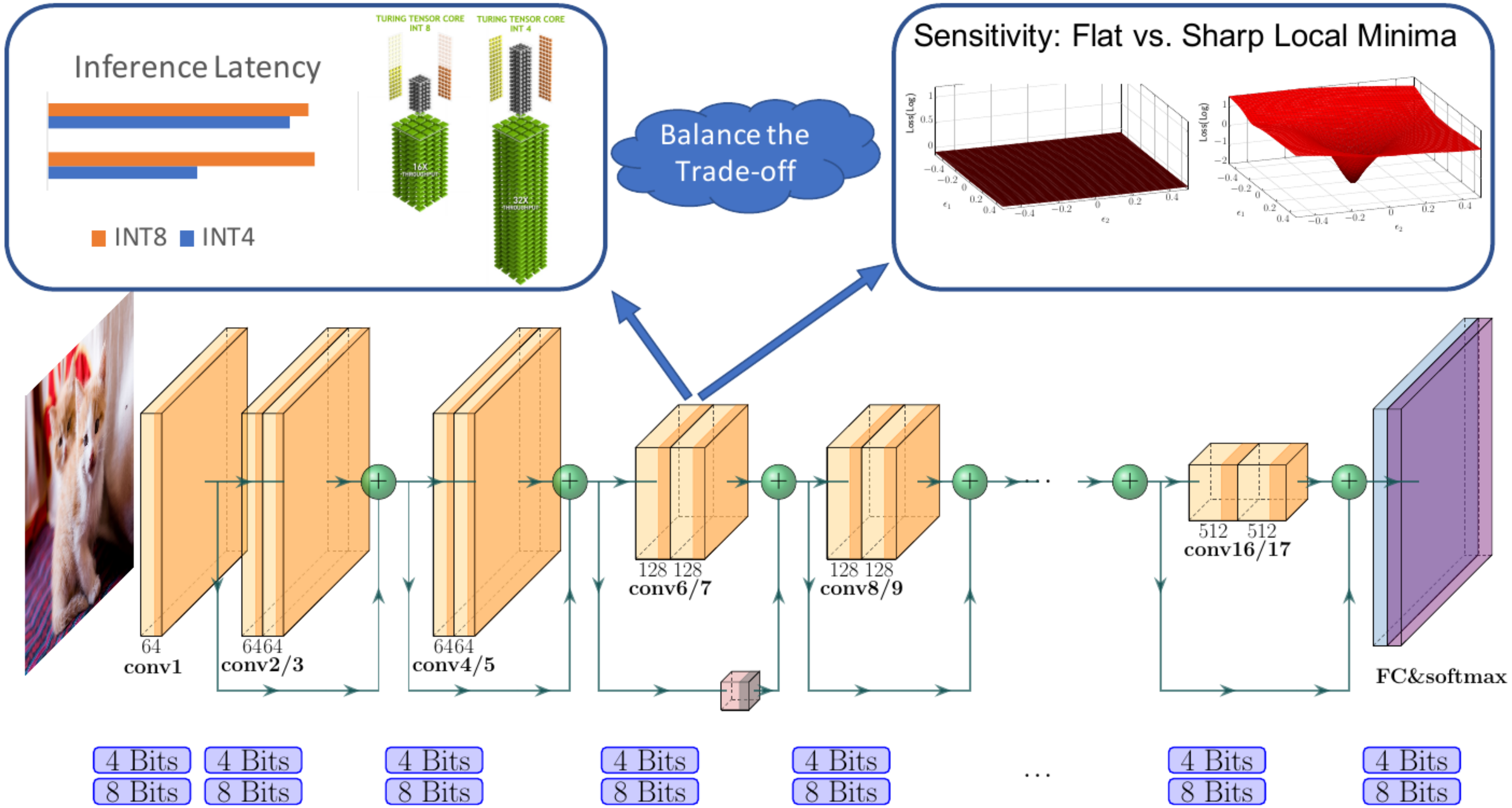
Simulated :
+ Better classification performances
- Does not benefit from low-precision logic

Integer-only :
+ Benefits from low-precision logic
- Lower classification performances
- Most works are limited to ReLU activations

Quantization

Mixed precision :

- Reinforcement learning approaches.
- NAS approaches.
- Regularization approaches.
- Hessian approaches.



HAWQ

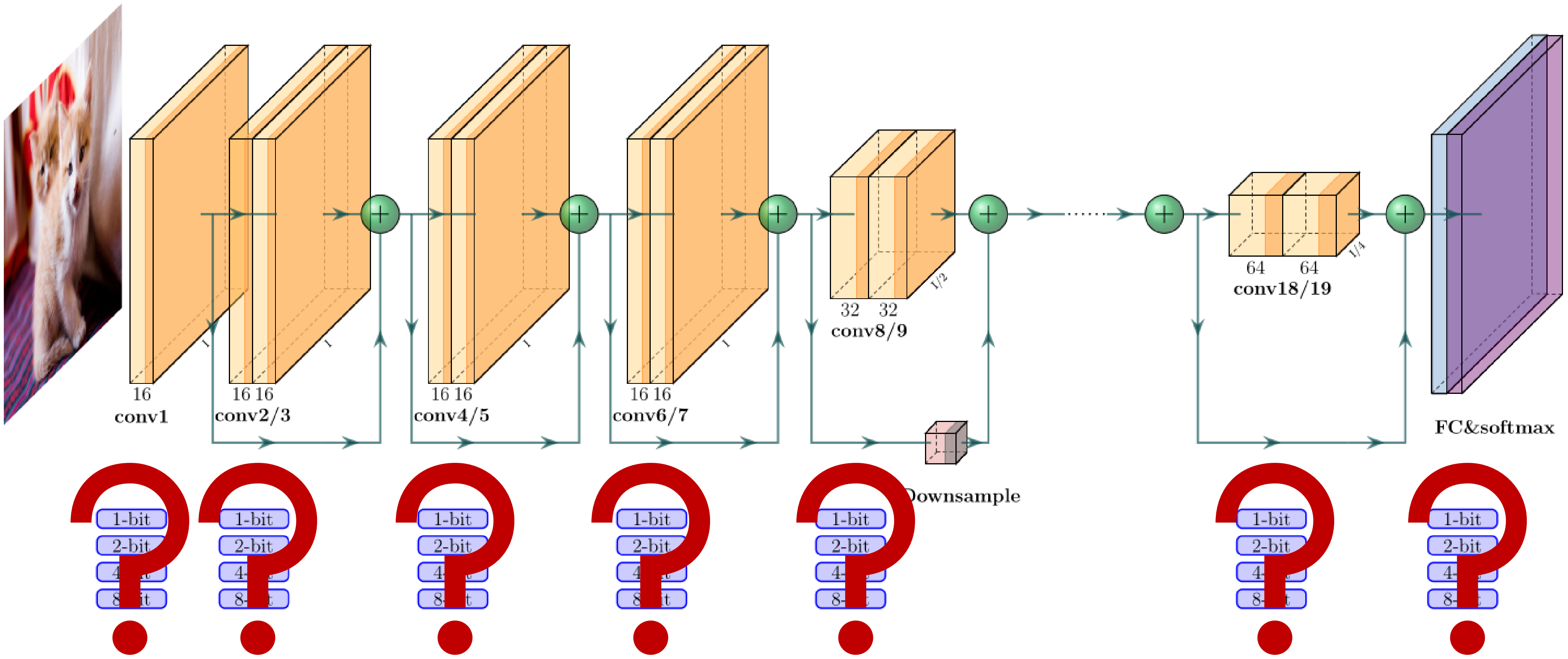


Figure – Hessian aware trace weighted quantization (HAWQ) for mixed quantization (**Dong et al. 2020**)

HAWQ

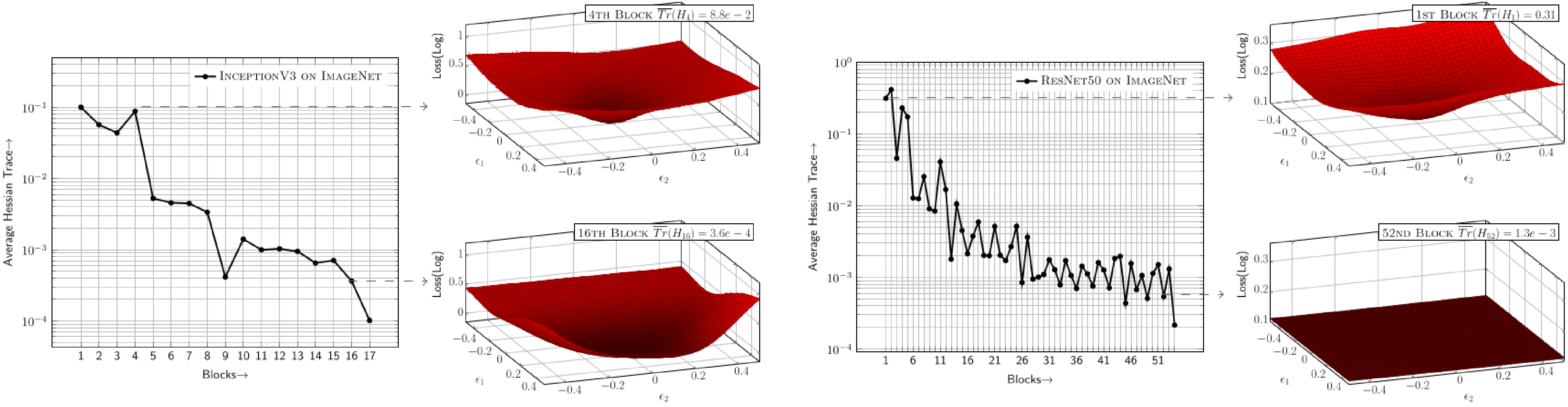


Figure – Hessian aware trace weighted quantization (HAWQ) for mixed quantization (**Dong et al. 2020**)

Ternary Weight Networks and DoReFa-Net

w → Full precision weight tensor

Ternary Weights Network

Binary Weights DoReFa-Net

$$w_l^t = \begin{cases} W_l : \tilde{w}_l > \Delta_l \\ 0 : |\tilde{w}_l| \leq \Delta_l \\ -W_l : \tilde{w}_l < -\Delta_l \end{cases}$$

$$w^b = \boldsymbol{E}(|\tilde{w}|) \times \operatorname{sign}(\tilde{w})$$

$$\Delta_l = 0.7 \times \boldsymbol{E}(|\tilde{w}_l|)$$

$$W_l = \underset{i \in \{i | \tilde{w}_l(i)| > \Delta\}}{\boldsymbol{E}} (|\tilde{w}_l(i)|)$$

Compression evaluation metrics

Sparsity-based metric

Full precision model

Quantized model

Function counting the number of quantized weights having a value of 0

$$SRQW(\mathcal{M}_{FP}, \mathcal{M}_C) = \frac{nzqw(\mathcal{M}_C)}{nqw(\mathcal{M}_{FP})}$$

Function counting the number of weights that can be quantized

Compression-based metrics

Function counting the number of bits necessary to store the weights (using COO sparse storage format)

$$CR(\mathcal{M}_{FP}, \mathcal{M}_Q) = \frac{nbits(\mathcal{M}_Q)}{nbits(\mathcal{M}_{FP})} \; ; \; CR_G(\mathcal{M}_{FP}, \mathcal{M}_Q) = 1 - CR(\mathcal{M}_{FP}, \mathcal{M}_Q)$$

Energy consumption evaluation metrics

Energy of mult-adds

Number of nonzero mult-adds

$$EC_{MA}(\mathcal{M}) = N_{MA} \times 3.7 \times 10^{-12}$$

Order of magnitude of the energy cost of a 32-bit multiplication (Horowitz et al. 2014)

Energy consumption evaluation metrics

Energy of data transfers

Number of layers in the model

Current layer

Number of bits necessary to encode the weights of the layer

$$EC_{DT}(\mathcal{M}) = 10^{-9} \times \sum_{i=1}^{p} \left(\left\lceil \frac{nnzw(L_i) \times B_i}{32} \right\rceil + N_{SF}^{i} \right)$$

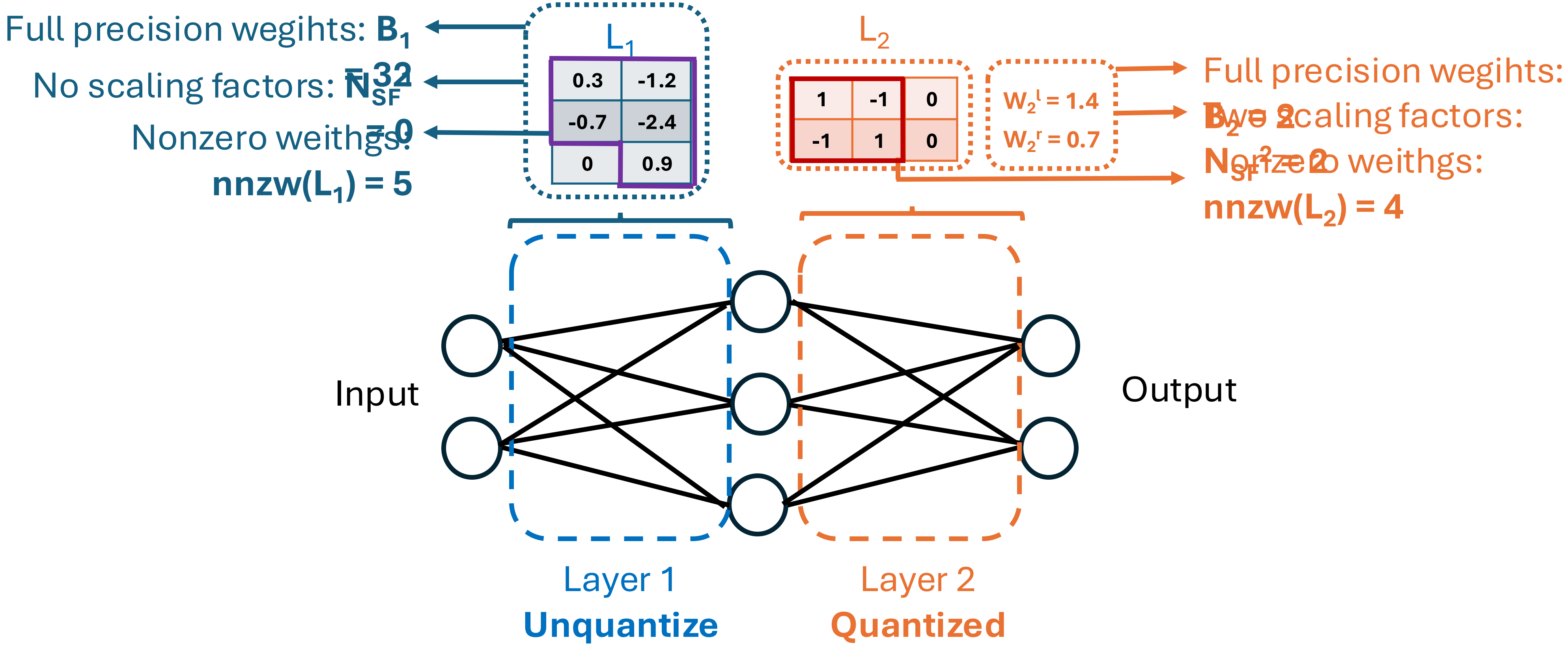
Order of magnitude of data transfers to memory from Molka et al. (2010)

Number of scaling factors

Function counting the number of nonzero weights

Energy consumption evaluation metrics

Energy of data transfers



$$EC_{DT}(\mathcal{M}) = 10^{-9} \times \left[\left(\left\lceil \frac{nnzw(L_1) \times B_1}{32} \right\rceil + N_{SF}^1\right) + \left(\left\lceil \frac{nnzw(L_2) \times B_2}{32} \right\rceil + N_{SF}^2\right)\right] = 10^{-9} \times \left[\left(\left\lceil \frac{5 \times 32}{32} \right\rceil + 0\right) + \left(\left\lceil \frac{4 \times 2}{32} \right\rceil + 2\right)\right]$$

$$= 10^{-9} \times [(5+0) + (1+2)] = 8 \times 10^{-9} \ J$$

Compression evaluation metrics

Compression-based metrics

Full precision (FP) model

Quantized model

Function counting the number of bits necessary to store the weights (using COO sparse storage format)

$$CR(\mathcal{M}_{FP}, \mathcal{M}_Q) = \frac{nbits(\mathcal{M}_Q)}{nbits(\mathcal{M}_{FP})}$$

$$CR_G(\mathcal{M}_{FP}, \mathcal{M}_Q) = 1 - CR(\mathcal{M}_{FP}, \mathcal{M}_Q)$$

The higher the better

Energy evaluation metrics

Energy consumption

- Energy consumption due to **Multiplications and Additions (MA)**.
- Takes into account **sparsity**

$$EC_{Total}(\mathcal{M}) = EC_{MA}(\mathcal{M}) + EC_{DT}(\mathcal{M})$$ in Joules

- Energy consumption due to **Data Transfers (DT)**
- Takes into account **sparsity** AND **reduced precision**

$$EC_S^{Total}(\mathcal{M}_{FP}, \mathcal{M}_Q) = \frac{|EC_{Total}(\mathcal{M}_{FP}) - EC_{Total}(\mathcal{M}_Q)|}{EC_{Total}(\mathcal{M}_{FP})}$$

The higher the better

Energy consumption evaluation metrics

Total energy

Takes into account sparsity

$$EC_T(\mathcal{M}) = EC_{MA}(\mathcal{M}) + EC_{DT}(\mathcal{M})$$

Takes into account sparsity AND reduced precision

$$EC_G^T(\mathcal{M}_{FP}, \mathcal{M}_C) = \frac{|EC_T(\mathcal{M}_{FP}) - EC_T(\mathcal{M}_C)|}{EC_T(\mathcal{M}_{FP})}$$

Takes into account sparsity

The higher the better

Experiment: SOTA comparison

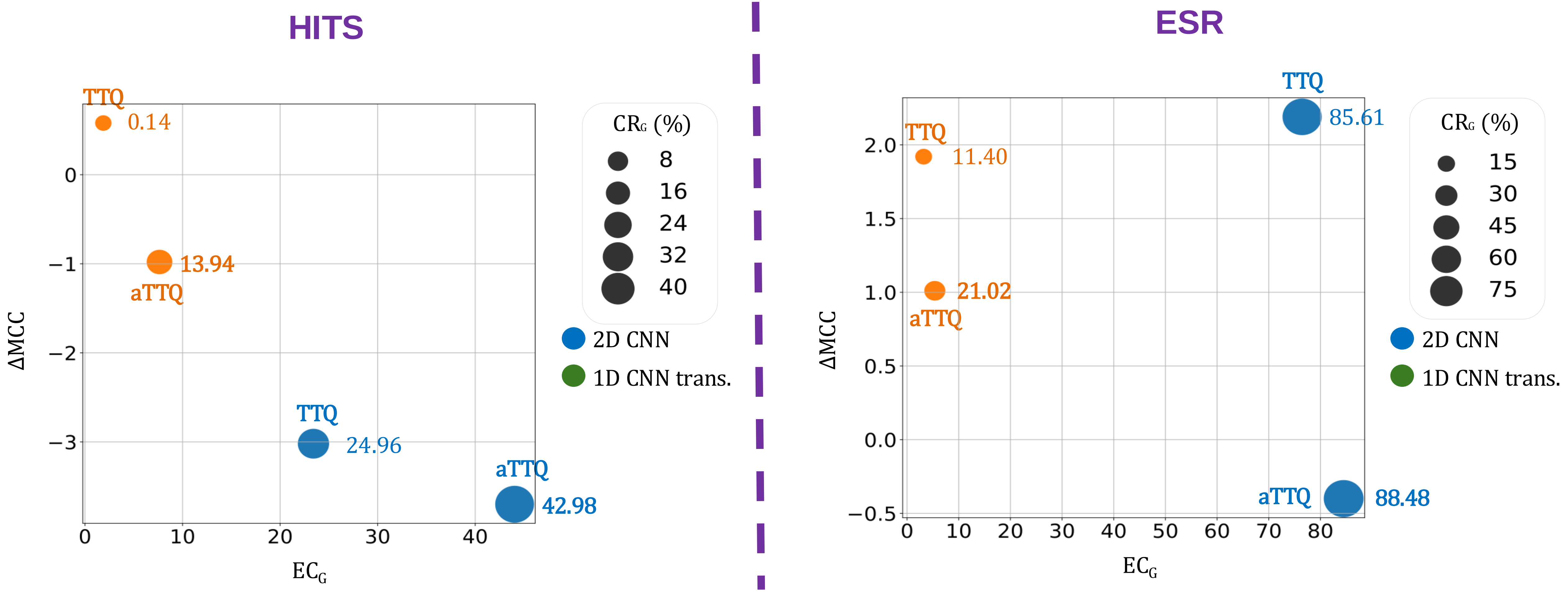
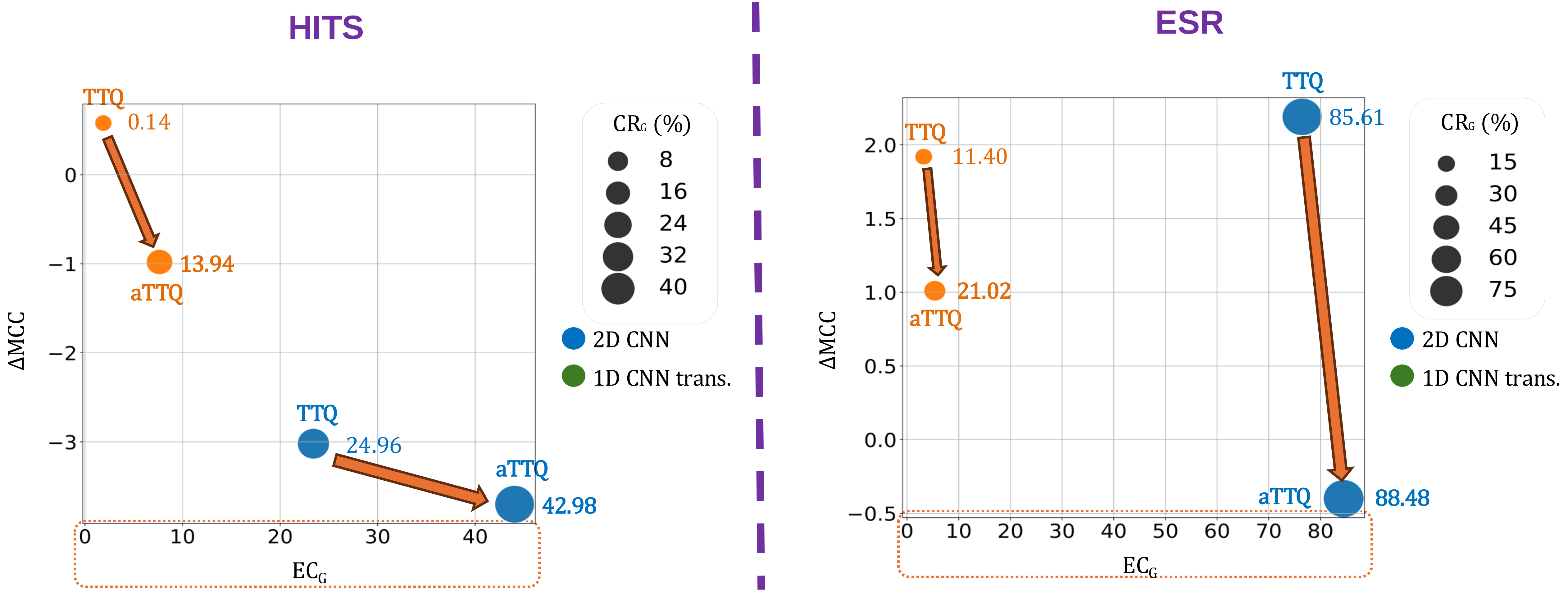


Figure – Comparison of aTTQ with FP and TTQ

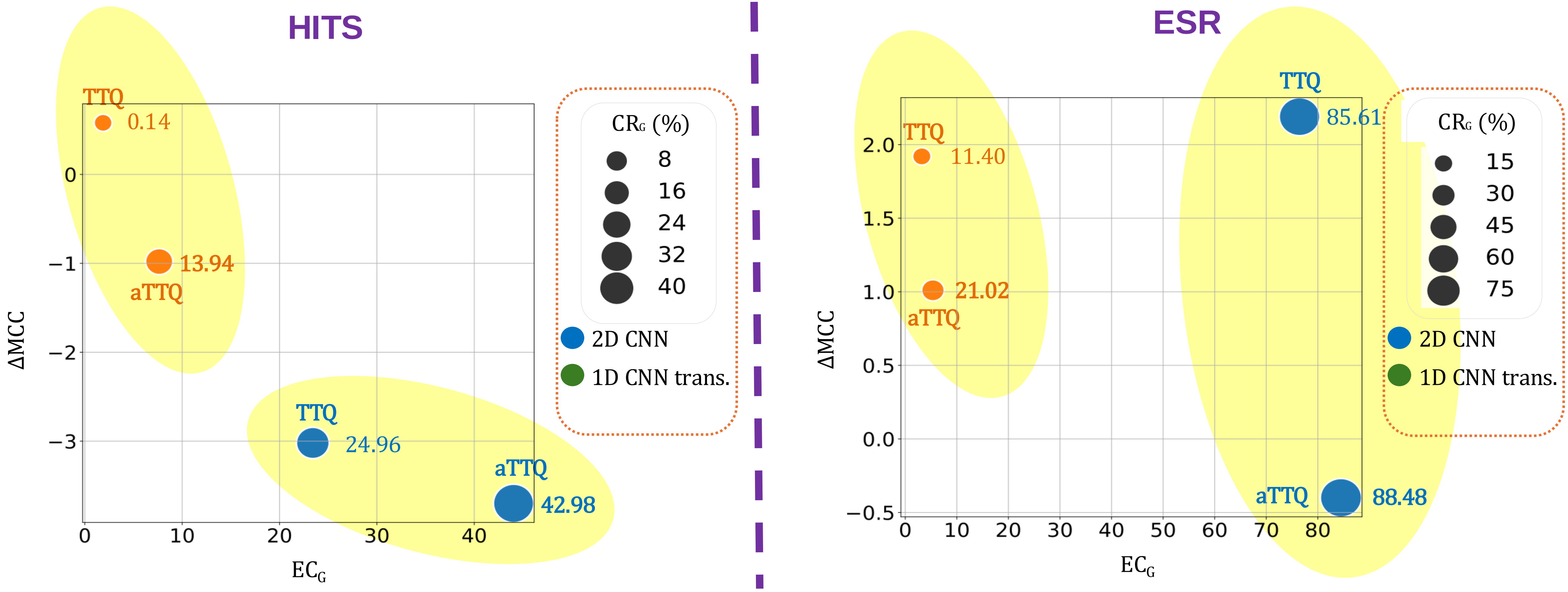
Experiment: SOTA comparison



aTTQ always have higher energy consumption gains compared to TTQ.

Figure – Comparison of aTTQ with FP and TTQ

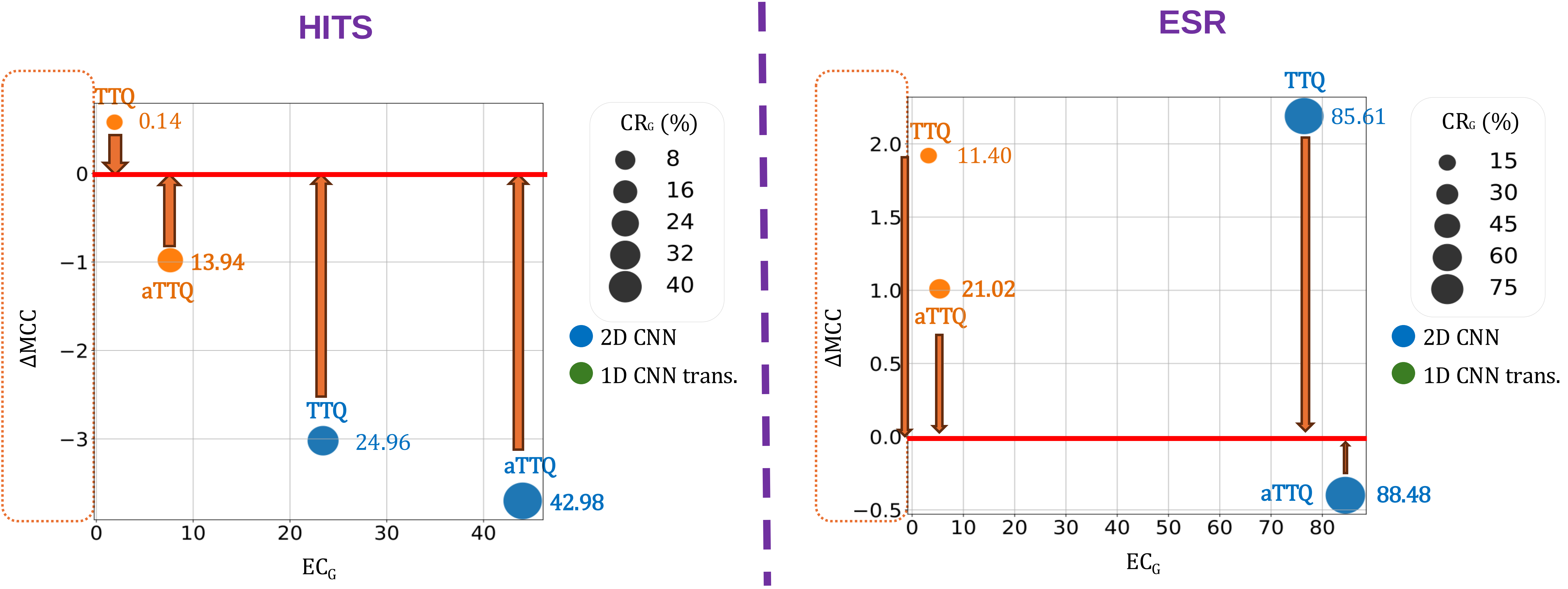
Experiment: SOTA comparison



aTTQ always achieves higher sparsity rates compared to TTQ.

Figure – Comparison of aTTQ with FP and TTQ

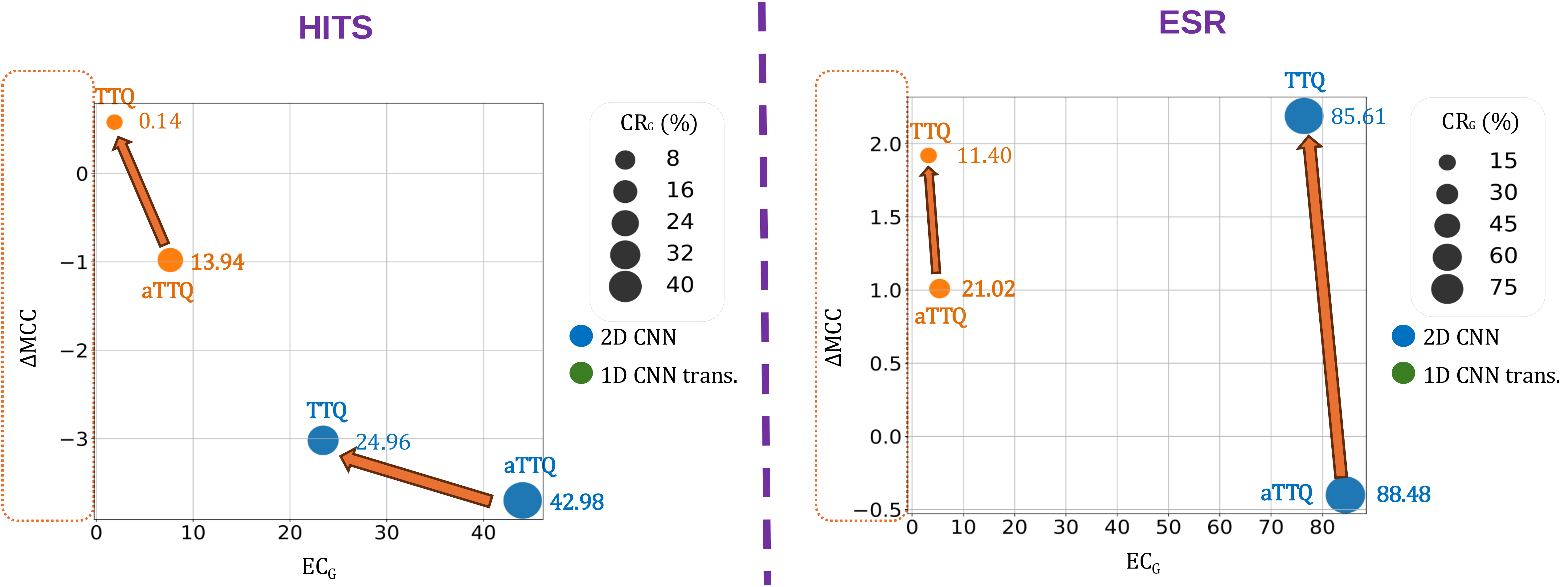
Experiment: SOTA comparison



Both aTTQ and TTQ perform similarly than the FP model in terms of classification

Figure – Comparison of aTTQ with FP and TTQ

Experiment: SOTA comparison



TTQ tend to have slightly higher classification performances than aTTQ

Figure – Comparison of aTTQ with FP and TTQ

Experiment: SOTA comparison

Dataset	Model	Quant. method	CR_G^T ↑	CR_G^Q ↑	$SRQW$ ↑	$\mathbf{EC_G^T}$ ↑	MCC ↑	Δ MCC ↑
HITS	2D CNN	FP	-	-	-	-	89.84 ± 3.09	-
		DoReFa [21]	**89.18** ± 0	**96.87** ± 0	-	3.54 ± 0	85.05 ± 5.96	−4.79
		TTQ [16]	24.96 ± 2.25	27.12 ± 2.44	28.96 ± 2.12	*23.42 ± 1.30*	**86.82 ± 2.29**	**−3.02**
		aTTQ	*42.98 ± 0.23*	*46.69 ± 0.25*	**45.95 ± 0.21**	**44.04 ± 0.19**	*86.14 ± 3.37*	*−3.70*
	1D CNN-trans.	FP	-	-	-	-	82.64 ± 1.77	-
		DoReFa [21]	**14.50** ± 0	**96.87** ± 0	-	0.37 ± 0.03	**84.07 ± 3.11**	**+1.43**
		TTQ [16]	0.14 ± 0.04	0.91 ± 0.27	6.75 ± 0.26	*1.88 ± 0.03*	*83.22 ± 2.36*	*+0.58*
		aTTQ	*13.94 ± 0.02*	*93.17 ± 0.16*	**93.53 ± 0.15**	**7.64 ± 0.11**	81.66 ± 4.17	−0.98
ESR	2D CNN	FP	-	-	-	-	92.81 ± 3.53	-
		DoReFa [21]	**96.40** ± 0	**96.87** ± 0	-	29.90 ± 0	*94.12 ± 0.87*	*+1.31*
		TTQ [16]	85.61 ± 1.37	86.03 ± 1.37	86.59 ± 1.29	*76.45 ± 1.13*	**95.00 ± 1.11**	**+2.19**
		aTTQ	*88.48 ± 0.44*	*88.91 ± 0.45*	**89.30 ± 0.42**	**84.49 ± 0.33**	92.41 ± 2.22	−0.40
	1D CNN-trans.	FP	-	-	-	-	94.33 ± 1.51	-
		DoReFa [21]	**23.46** ± 0	**96.86** ± 0	-	0.90 ± 0	**96.79 ± 0.55**	**+2.46**
		TTQ [16]	11.40 ± 2.61	47.07 ± 10.79	50.22 ± 10.16	*3.21 ± 0.66*	*96.25 ± 0.79*	*+1.92*
		aTTQ	*21.02 ± 0.15*	*86.78 ± 0.63*	**87.59 ± 0.59**	**5.37 ± 0.04**	95.34 ± 0.79	+1.01
MNIST	2D MNIST CNN	FP	-	-	-	-	94.39 ± 0.46	-
		DoReFa [21]	**51.67** ± 0	**96.84** ± 0	-	*3.28 ± 0*	87.03 ± 7.14	−7.36
		TTQ [16]	13.86 ± 2.33	25.97 ± 4.37	30.40 ± 4.12	2.58 ± 0.35	*92.09 ± 0.89*	*−2.30*
		aTTQ	*28.98 ± 1.26*	*54.32 ± 2.36*	**57.08 ± 2.22**	**4.97 ± 0.22**	**93.62 ± 0.96**	**−0.77**

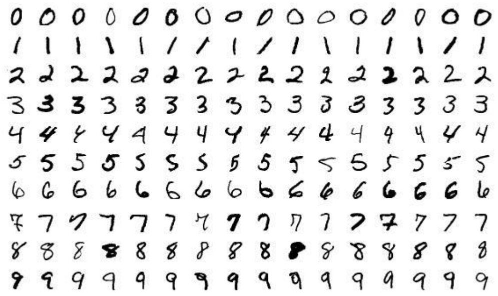
Table – Comparison of **aTTQ** with other quantization methods

Experiment: influence of normalization

Objective:

- Study the influence of normalization on aTTQ.

Datasets:

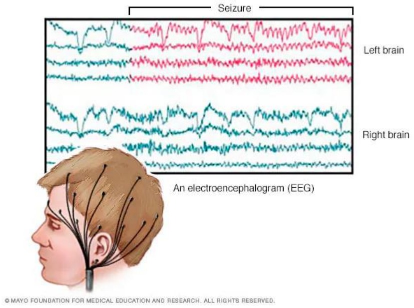


MNIST subset:

- 28x28 images.
- 20 000 samples.
- Ten classes.

HITS:

- TCD Data.
- 1 545 samples.
- Three classes.
- Sampling frequency: 4385 Hz.



ESR:

- EEG Data.
- 11 500 samples.
- Two classes.
- Sampling frequency: 174 Hz.

Metrics:

- Mathews Correlation Coefficient (MCC).
- CR_G.

Models:

- 2D CNN.
- 1D CNN-transformer.

Loss function:

- Cross entropy (CE).

Experiment: influence of normalization

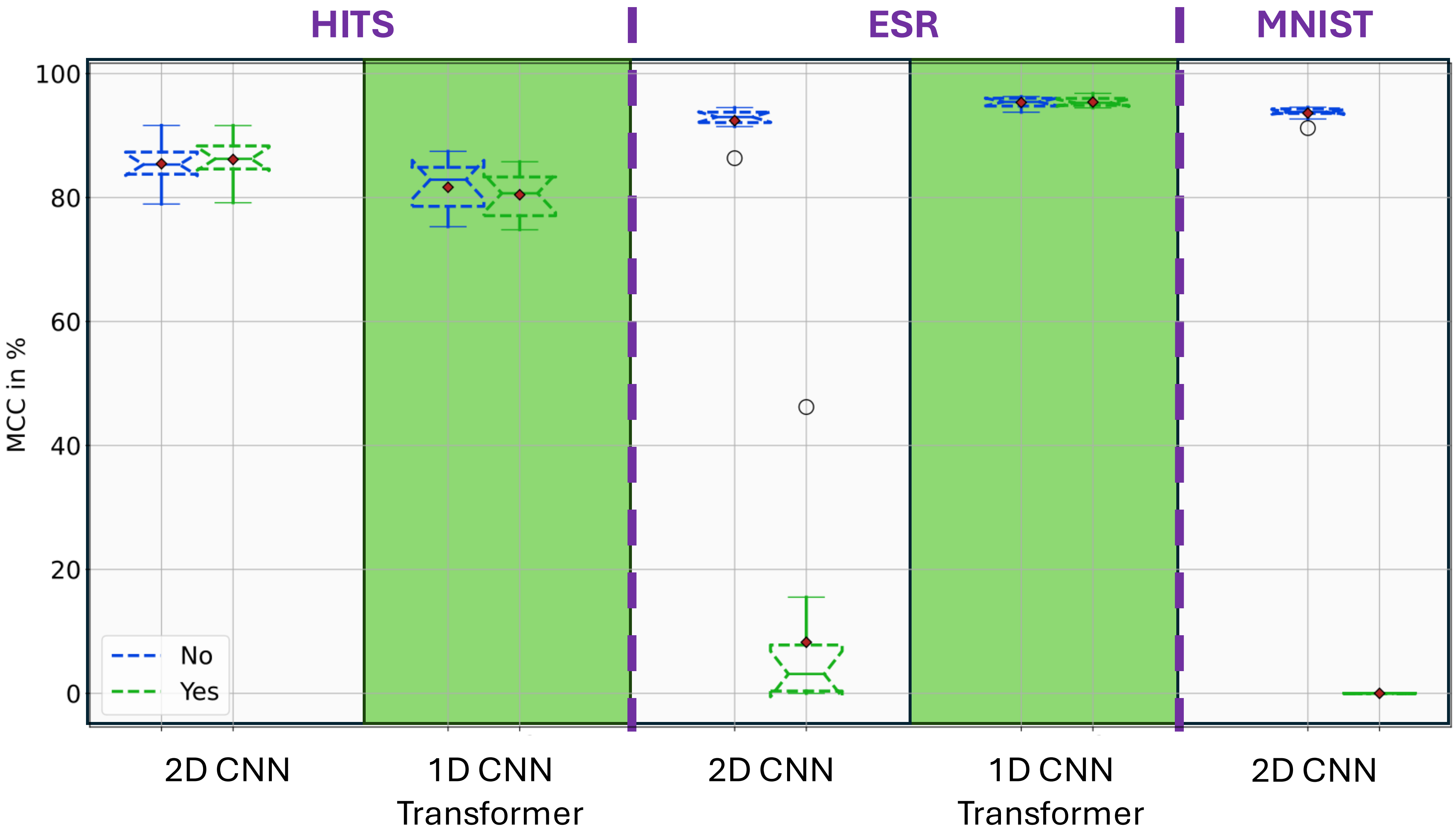


Figure – Influence of normalization on aTTQ from the **classification** perspective

Experiment: influence of normalization

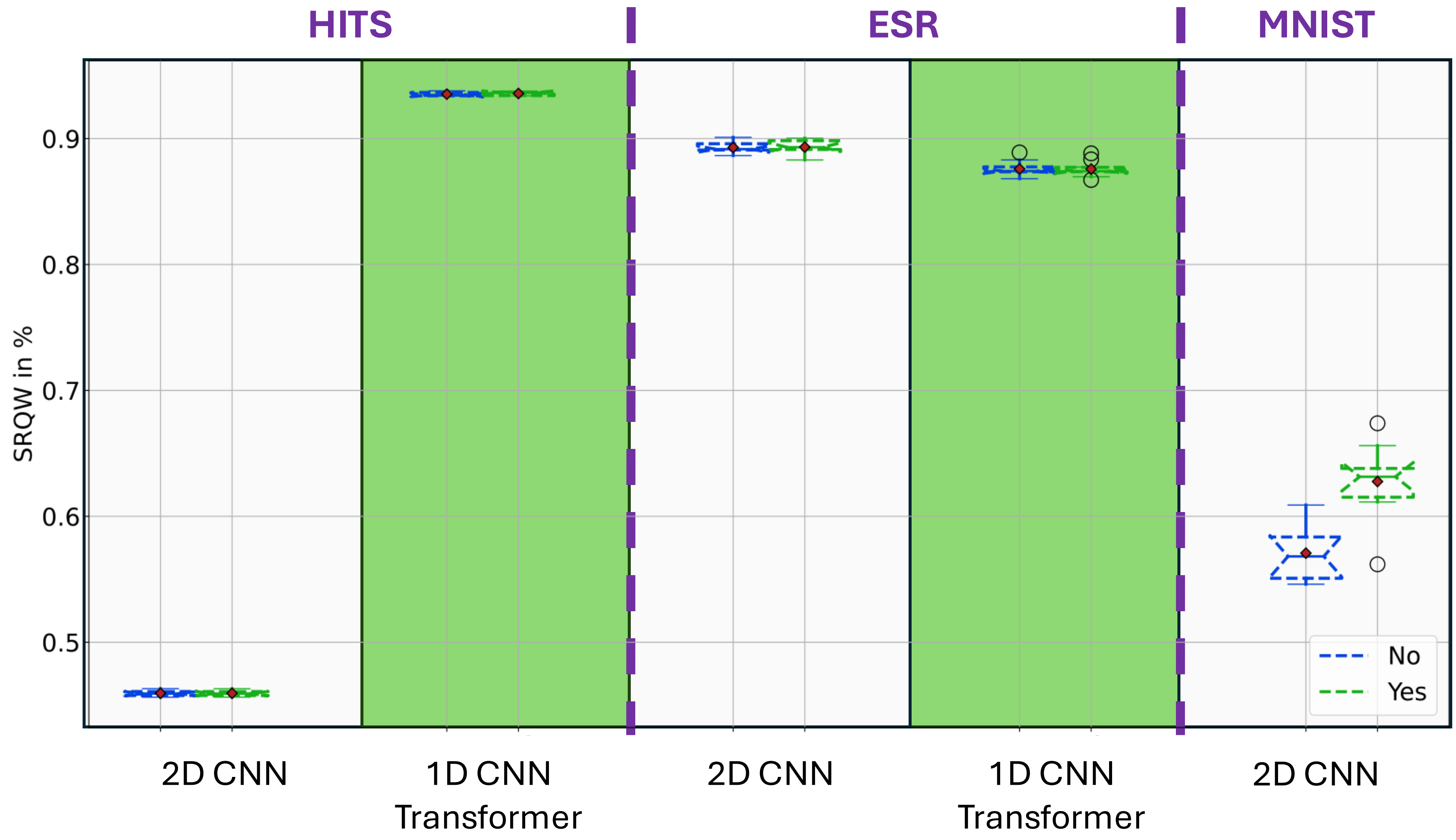
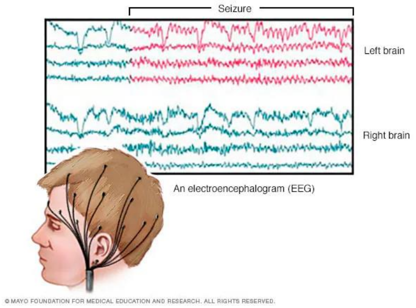


Figure – Influence of normalization on aTTQ from the **sparsity/compression** perspective

Experiment: influence of t_{min} and t_{max}

Objective:
- Study the influence of our trade-off parametrization t_{min} and t_{max}.

Dataset:



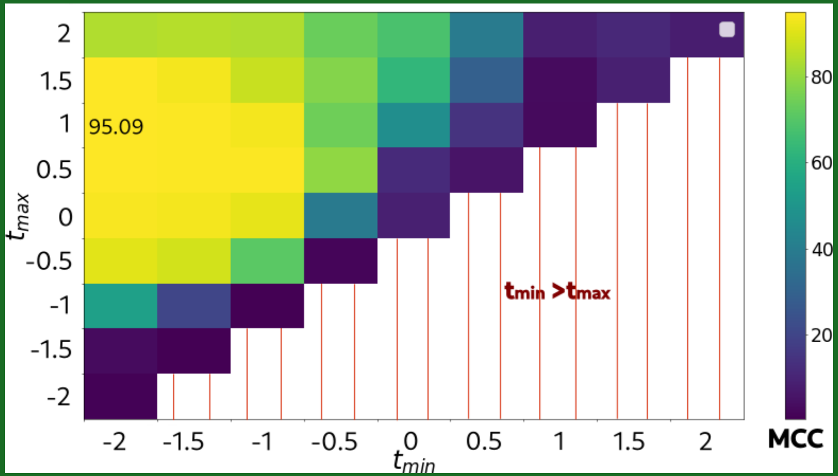
ESR:
- EEG Data.
- 11 500 samples.
- Two classes.
- Sampling frequency: 174 Hz.

Metrics:
- Matthews correlation coefficient (MCC).
- Sparsity rate of the quantized weights (SRQW).

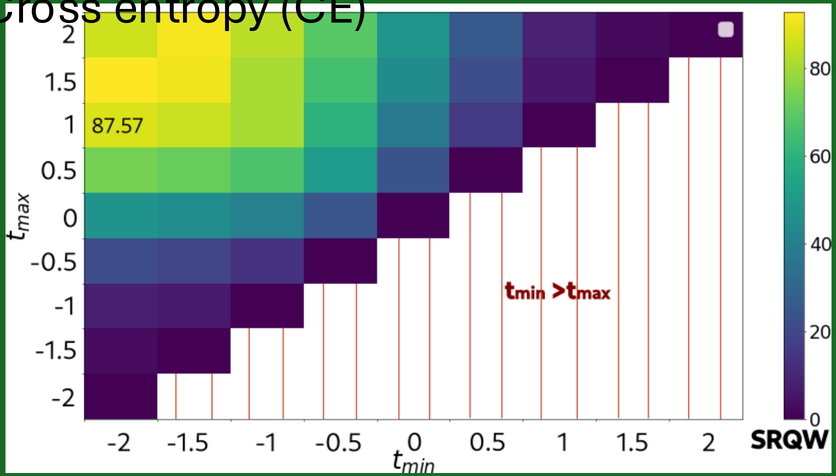
Model:
- 1D CNN-transformer.

Loss function:
- Cross entropy (CE)

ESR 1D CNN-transformer



MCC



SRQW

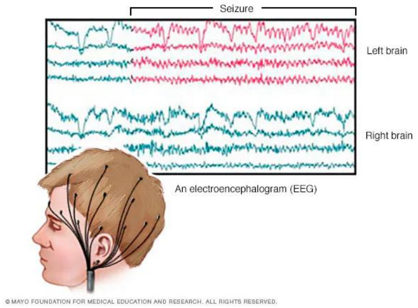
Figure – SMCC and SRQW for different values of t_{min} and t_{max}.

Experiment: influence of t_{min} and t_{max}

Objective:

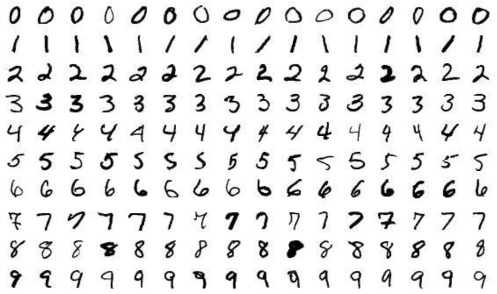
- Study the influence of our trade-off parametrization t_{min} and t_{max}.

Datasets:



ESR:

- EEG Data.
- 11 500 samples.
- Two classes.
- Sampling frequency: 174 Hz.



MNIST subset:

- 28x28 images.
- 20 000 samples.
- Ten classes.

Metrics:

- Mathews Correlation Coefficient (MCC).
- CR_G.

Models:

- 2D CNN.
- 1D CNN-transformer.

Loss function:

- Cross entropy (CE)

Experiment: influence of t_{min} and t_{max}

MNIST 2D CNN

ESR 1D CNN-transformer

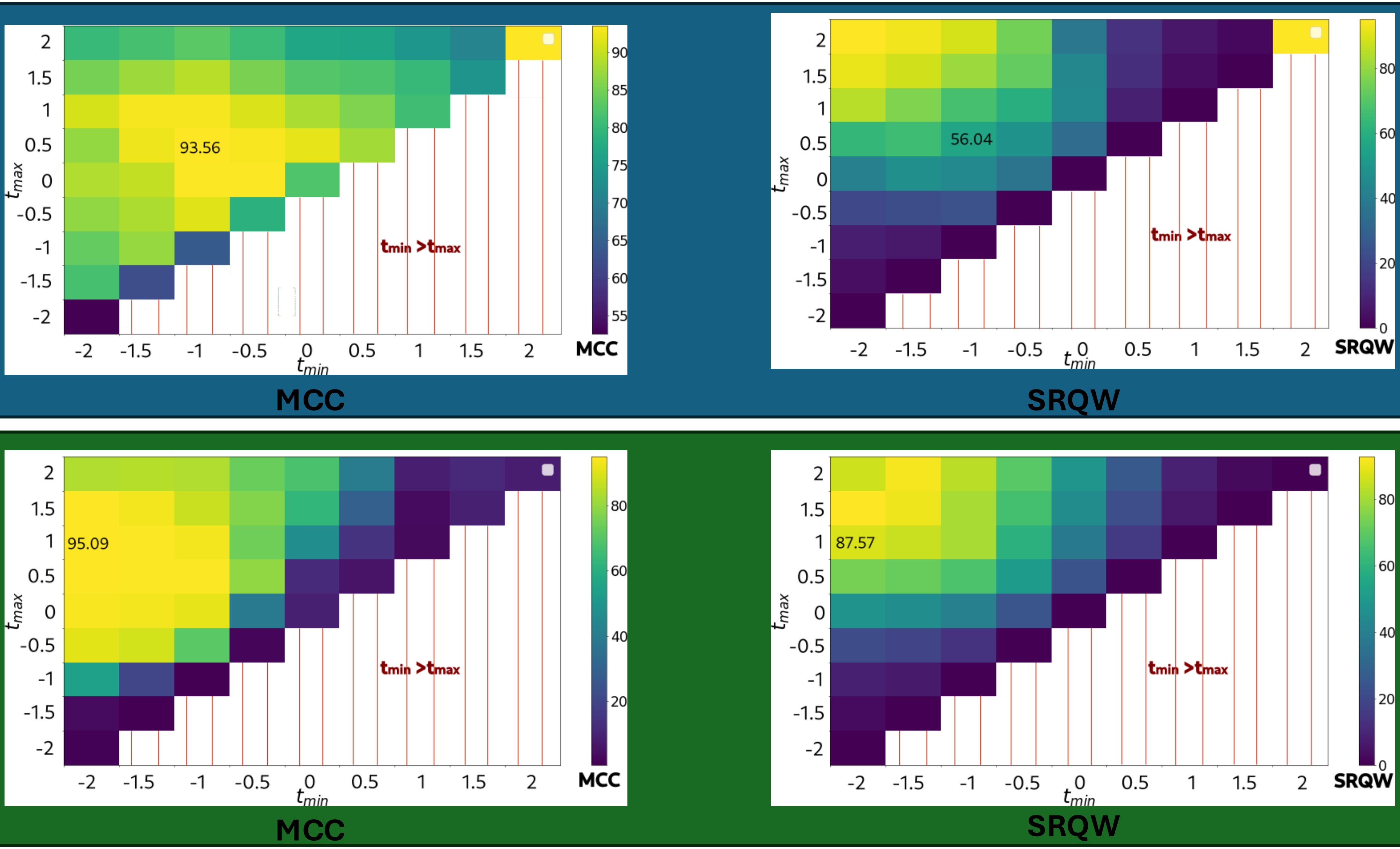
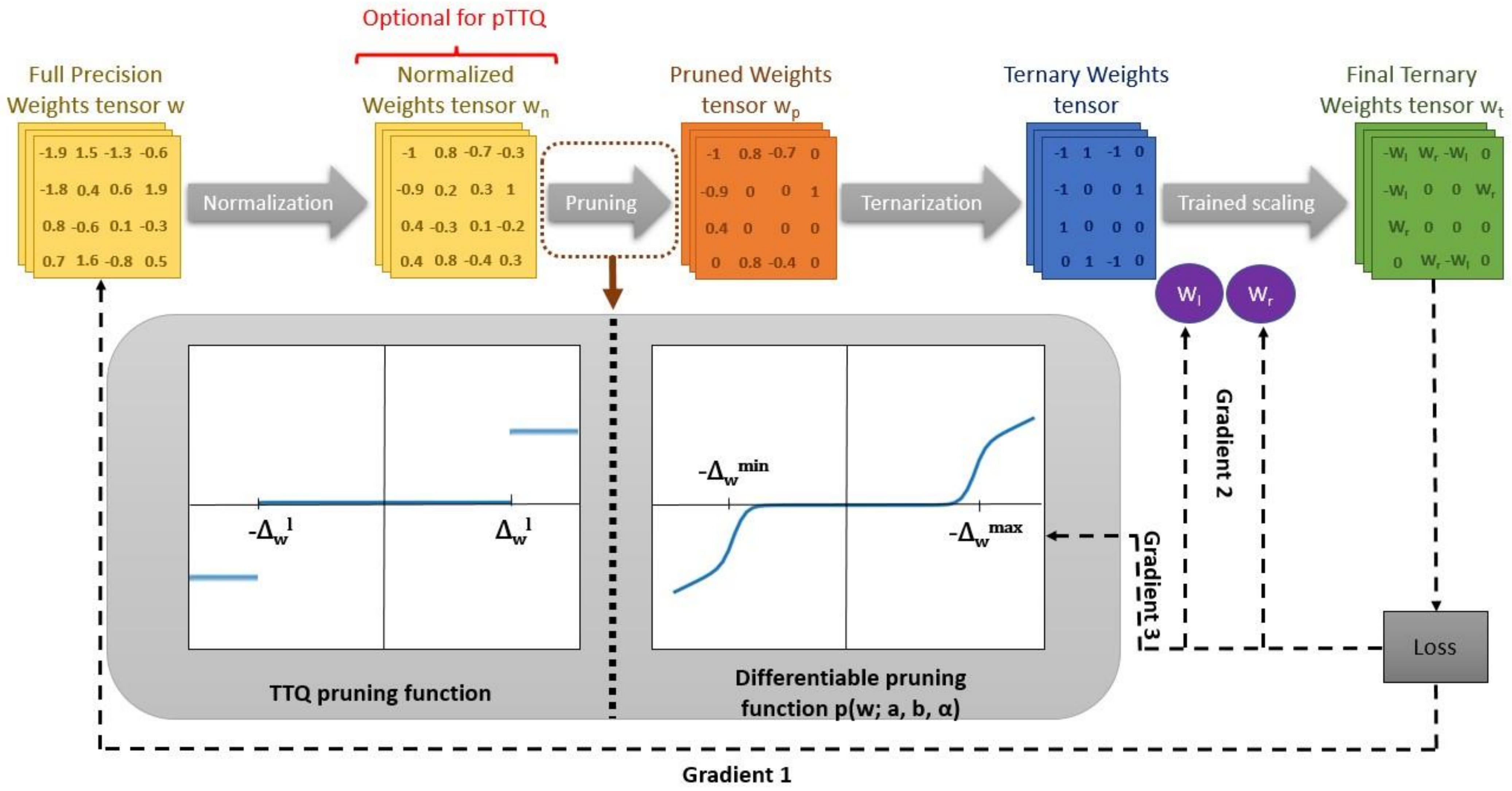


Figure – SMCC and SRQW for different values of t_{min} and t_{max}.

Pruned trained ternary quantization (pTTQ)



Pruned trained ternary quantization (pTTQ)

Pruning function:

Slope of thresholding

$$p(w; t_{min}, t_{max}, \alpha) = ReLU(w - \Delta_w(t_{max})) + \Delta_w(t_{max}) * S(\alpha \times (w - \Delta_w(t_{max})))$$

$$- ReLU(-w - \Delta_w(t_{min})) - \Delta_w(t_{min}) * S(\alpha \times (-w - \Delta_w(t_{min})))$$

$$\Delta_w : t \to \mu_w + t \times \sigma_w$$

Sigmoid function

Mean

Standard deviation

Pruned trained ternary quantization (pTTQ)

Gradients:

Heaviside function

$$\frac{\partial p}{\partial t_{min}}(w; t_{min}, t_{max}, \alpha) = \sigma_w \times H(-w - \Delta_w^{min}) - \sigma_w \times S(\alpha \times (-w - \Delta_w^{min}))$$
$$+ \sigma_w \times \alpha \times \Delta_w^{min} \times S(\alpha \times (-w - \Delta_w^{min})) \times (1 - S(-w - \Delta_w^{min}))$$

$$\frac{\partial p}{\partial t_{min}}(w; t_{min}, t_{max}, \alpha) = -\sigma_w \times H(w - \Delta_w^{max}) + \sigma_w \times S(\alpha \times (w - \Delta_w^{max}))$$
$$- \sigma_w \times \alpha \times \Delta_w^{max} \times S(\alpha \times (w - \Delta_w^{max})) \times (1 - S(w - \Delta_w^{max}))$$

Pruned trained ternary quantization (pTTQ)

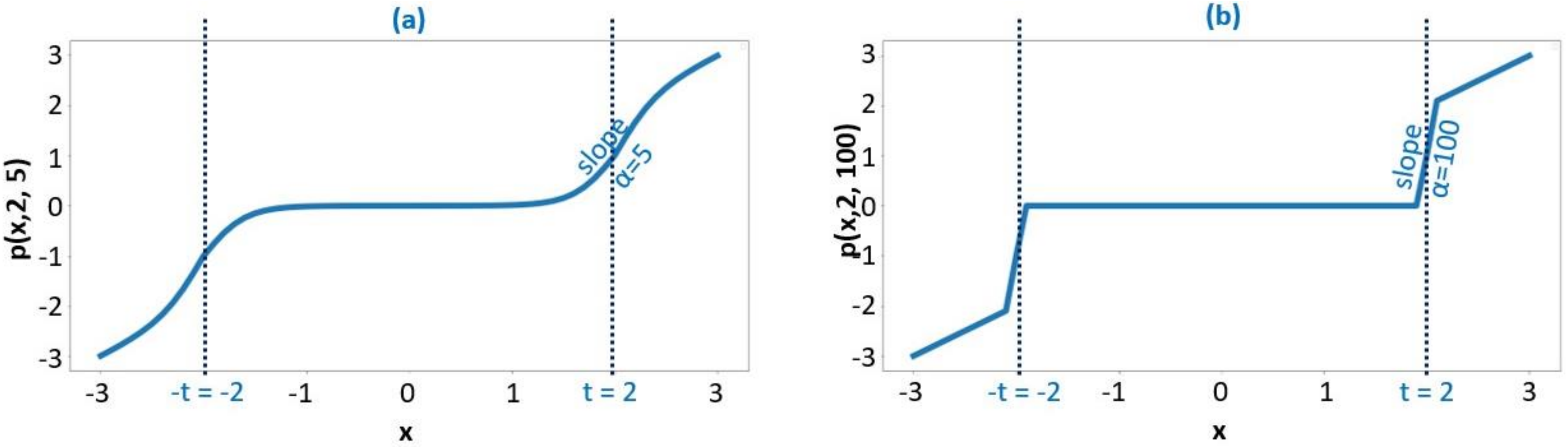


Figure – Examples of pruning functions for different thresholds and values of α

Pruned trained ternary quantization (pTTQ)

Dataset	Model	Quant. method	$CR_G^T \uparrow$	$CR_G^Q \uparrow$	$SRQW$	$EC_G^T \uparrow$	MCC ↑	Δ MCC ↑
HITS	2D CNN	FP	-	-	-	-	89.84 ± 3.09	-
		DoReFa [33]	**89.18 ± 0**	**96.87 ± 0**	-	3.54 ± 0	85.05 ± 5.96	−4.79
		TTQ [13]	24.96 ± 2.25	27.12 ± 2.44	28.96 ± 2.12	*23.42 ± 1.30*	*86.82 ± 2.29*	*−3.02*
		pTTQ	*75.54 ± 3.39*	*82.06 ± 3.69*	**83.12 ± 3.47**	**75.53 ± 1.53**	**89.33 ± 4.45**	**−0.55**
	1D CNN-trans.	FP	-	-	-	-	82.64 ± 1.77	-
		DoReFa [33]	**14.50 ± 0**	**96.87 ± 0**	-	0.37 ± 0.03	*84.07 ± 3.11*	*+1.43*
		TTQ [13]	0.14 ± 0.04	0.91 ± 0.27	6.75 ± 0.26	*1.88 ± 0.03*	83.22 ± 2.36	+0.58
		pTTQ	*8.37 ± 0.05*	*55.89 ± 0.34*	**58.50 ± 0.32**	**2.01 ± 0.05**	**85.12 ± 1.94**	**+2.48**
ESR	2D CNN	FP	-	-	-	-	92.81 ± 3.53	-
		DoReFa [33]	**96.40 ± 0**	**96.87 ± 0**	-	29.90 ± 0	*94.12 ± 0.87*	*+1.31*
		TTQ [13]	85.61 ± 1.37	86.03 ± 1.37	86.59 ± 1.29	*76.45 ± 1.13*	**95.00 ± 1.11**	*+2.19*
		pTTQ	*93.35 ± 0.96*	*93.80 ± 0.96*	**94.17 ± 0.91**	**90.32 ± 0.69**	92.23 ± 2.32	−0.58
	1D CNN-trans.	FP	-	-	-	-	94.33 ± 1.51	-
		DoReFa [33]	*23.46 ± 0*	*96.86 ± 0*	-	0.90 ± 0	**96.79 ± 0.55**	**+2.46**
		TTQ [13]	11.40 ± 2.61	47.07 ± 10.79	50.22 ± 10.16	*3.21 ± 0.66*	96.25 ± 0.79	+1.92
		pTTQ	**23.86 ± 0.04**	**98.54 ± 0.16**	**98.67 ± 0.15**	**6.04 ± 0.01**	*96.35 ± 0.95*	*+2.02*
MNIST	2D MNIST CNN	-	-	-	-	-	94.39 ± 0.46	-
		DoReFa [33]	**51.67 ± 0**	**96.84 ± 0**	-	*3.28 ± 0*	87.03 ± 7.14	−7.36
		TTQ [13]	13.86 ± 2.33	25.97 ± 4.37	30.40 ± 4.12	2.58 ± 0.35	**92.09 ± 0.89**	**−2.30**
		pTTQ	*33.92 ± 1.02*	*63.58 ± 1.92*	**65.79 ± 1.80**	**6.10 ± 0.15**	*91.01 ± 0.61*	*−3.38*

Table – Comparison of pTTQ with other state-of-the-art methods

Conclusion et perspectives

Conclusion

Dataset creation and annotation

- Semi-supervised data annotation
- Soft labelling (annotation)

Novel methodology for **semi-automatic data annotation** based on local-quality metrics.

Selection strategy of the best **projection** obtained by a dimensionality reduction technique.

Use robust loss functions to improve the classification performances of a **classifier trained** on a **noisy** semi-automatic labeled **dataset.**

Conclusion

Multiple representations

- Different models with different inputs
- Multi-feature models

Novel **hybrid CNN-transformer** models, **exploiting** the **complementarity** between the **temporal** and **spectral characteristics** of a medical signal.

Guided and **regularized intermediate fusion** approach, improving generalization while handling **imbalanced** datasets and **label-noise**.

Late-fusion mechanisms, based on **learnable** and **interpretable attention weights**.

Conclusion

Resource hungry models →

- Lite models
- Model compression
- (Soft labelling training)

→ **Novel ternarization heuristic**, based on the weights' statistics.

→ **Direct asymmetric pruning** before ternarization, allowing a **better trade-off** between compression, energy, and classification.

→ **Asymmetric parametrization** of the sparsity rate, **controlling** the abovementioned **trade-off**.

Perspectives

Dataset creation and annotation

- Semi-supervised data annotation
- Soft labelling (annotation)

More complex encoding models (VAE, GANs, diffusion models, ...)

Stronger regularization (DEC, contrastive learning, more complex projection metrics, ...)

Active learning by proposing to human experts the most difficult samples

Perspectives

Multiple representations →

- Different models with different inputs
- Multi-feature models

Test other types of models (LSTM, ViT, ResNet, …) **and datasets** (medical and non-medical)

Use **other representations** of the raw signal (cochleagram, binary encodings, chromagram, …)

Use **other** types of **regularization** (contrastive learning with weak supervision, link constraints, …)

Perspectives

Resource hungry models →

- Lite models
- Model compression
- (Soft labelling training)

→ **Differentiable pruning function**, with asymmetric learnable parameters

→ **Mixed quantization** to completely quantize the models with different precisions

→ **Hardware implementation** to take advantage of the compressed models

Perspectives

Soft labelling

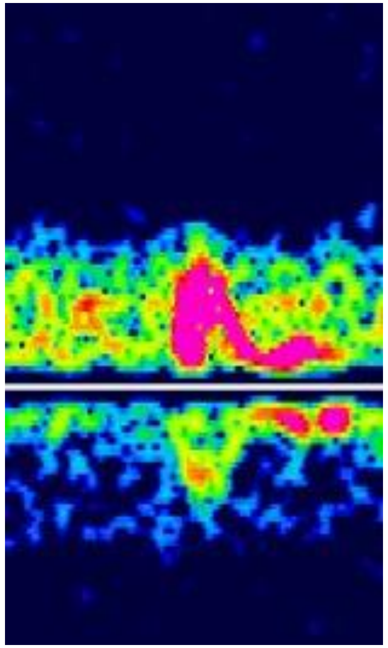
- Capture expert uncertainty
- Noise robustness

Take advantage of soft annotations using **soft-lables loss functions** (soft CE, Jensen-Shannon divergence, …), to capture the human expert uncertainty,

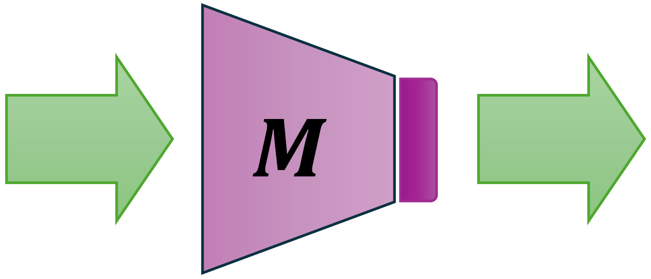
New **loss functions robust** against **soft-label noise** (geometric mean Jensen-Shannon divergence).

Semi-automatic soft label annotation evaluation.

Working with soft labels (vs hard labels)



Sample X



Deep model

Soft prediction

$$\boldsymbol{M}(\boldsymbol{X})_S = \begin{pmatrix} \mathbf{y}_s^{Art} \\ \mathbf{y}_s^{GE} \\ \mathbf{y}_s^{SE} \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.3 \\ \boxed{0.6} \end{pmatrix}$$

conversion to
hard prediction
→ highest score

$$\boldsymbol{M}(\boldsymbol{X})_H = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Soft labels

Loss functions

	soft labels	noise tolerant
$L_{CE}(y_H, M(X)_H) = H(y_H, M(X)_H)$ cross-entropy (baseline)	✖	✖
$L_{SoftCE}(y_S, M(X)_S) = H(y_S, M(X)_S)$	✔	✖
$L_{SymCE}(y_H, M(X)_H) = \alpha \times H(y_H, M(X)_H) + \beta \times H(M(X)_H, y_H)$	✖	✔
$L_{JSD}(y_S, M(X)_S) = \frac{1}{2} \times (KL(y_S, m_S) + KL(M(X)_S, m_S))$ with $m_S = \frac{1}{2} \times (y_S + M(X)_S)$	✔	✖

Adding symmetric noise to soft labels

noise rate ζ (in %)

$$y_S \begin{pmatrix} 0.1 \\ 0.3 \\ \boxed{0.6} \end{pmatrix} \begin{matrix} \text{Art} \\ \text{GE} \\ \text{SE} \end{matrix}$$

Highest score moves to another class with probability ζ

with probability $\frac{\zeta}{C-1}$ → $\overline{y_S} = \begin{pmatrix} \boxed{0.6} \\ 0.3 \\ 0.1 \end{pmatrix}$ **Noisy**

with probability $\frac{\zeta}{C-1}$ → $\overline{y_S} = \begin{pmatrix} 0.1 \\ \boxed{0.6} \\ 0.3 \end{pmatrix}$ **Noisy**

with probability $1 - \zeta$ → $\overline{y_S} = \begin{pmatrix} 0.1 \\ 0.3 \\ \boxed{0.6} \end{pmatrix}$ **Unmodified**

Adding symmetric noise to soft labels

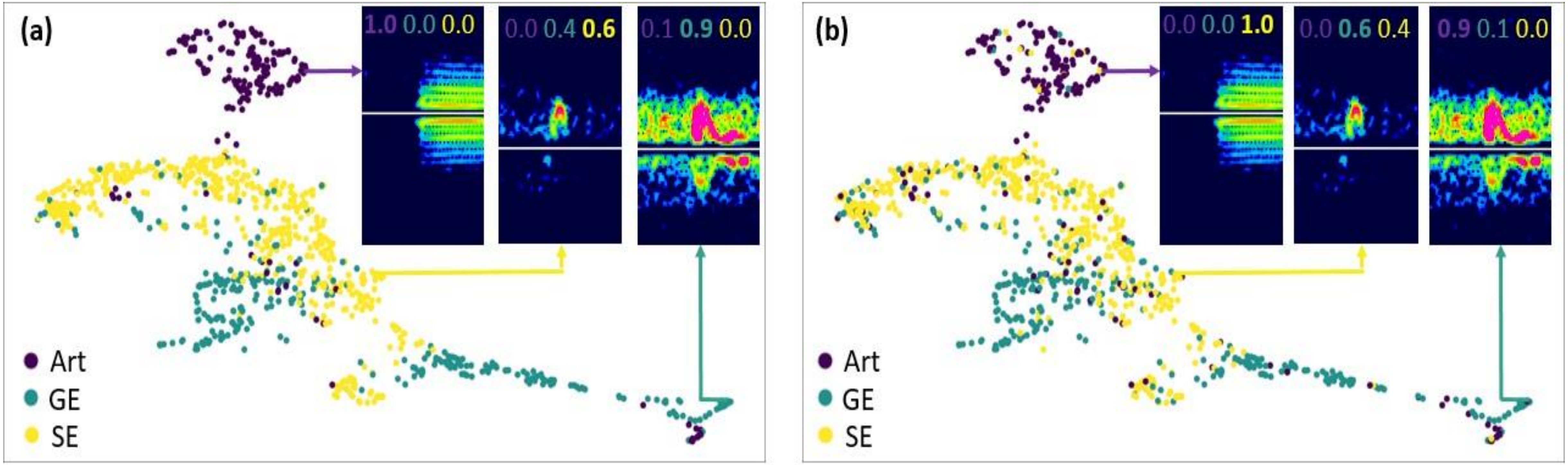
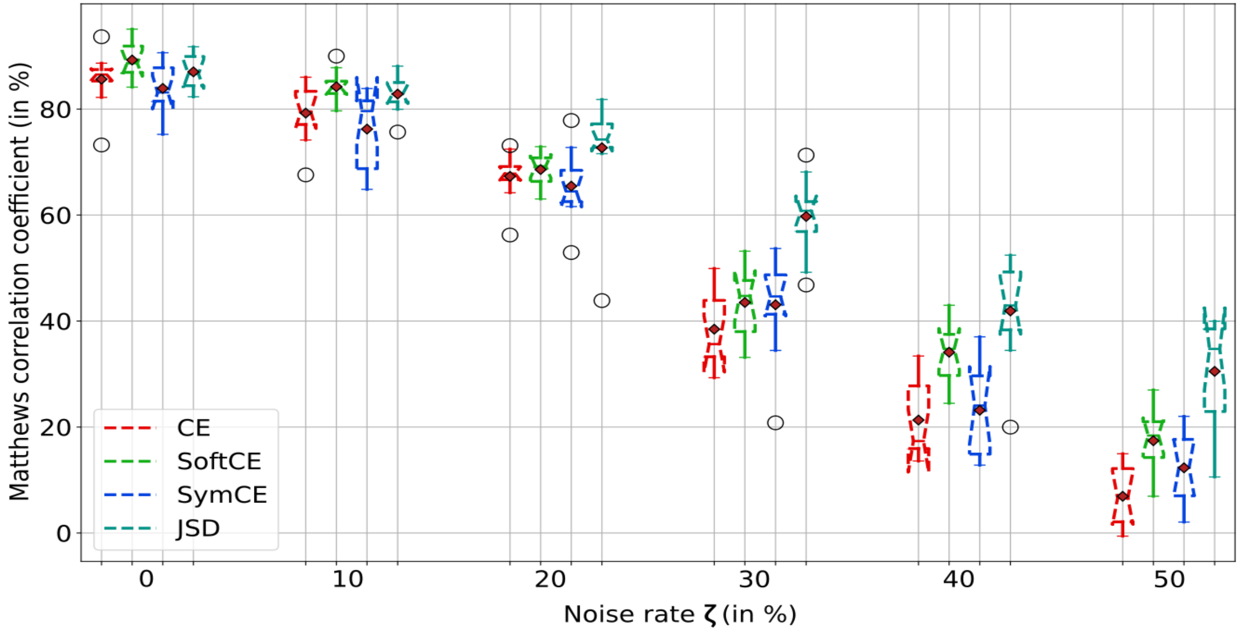
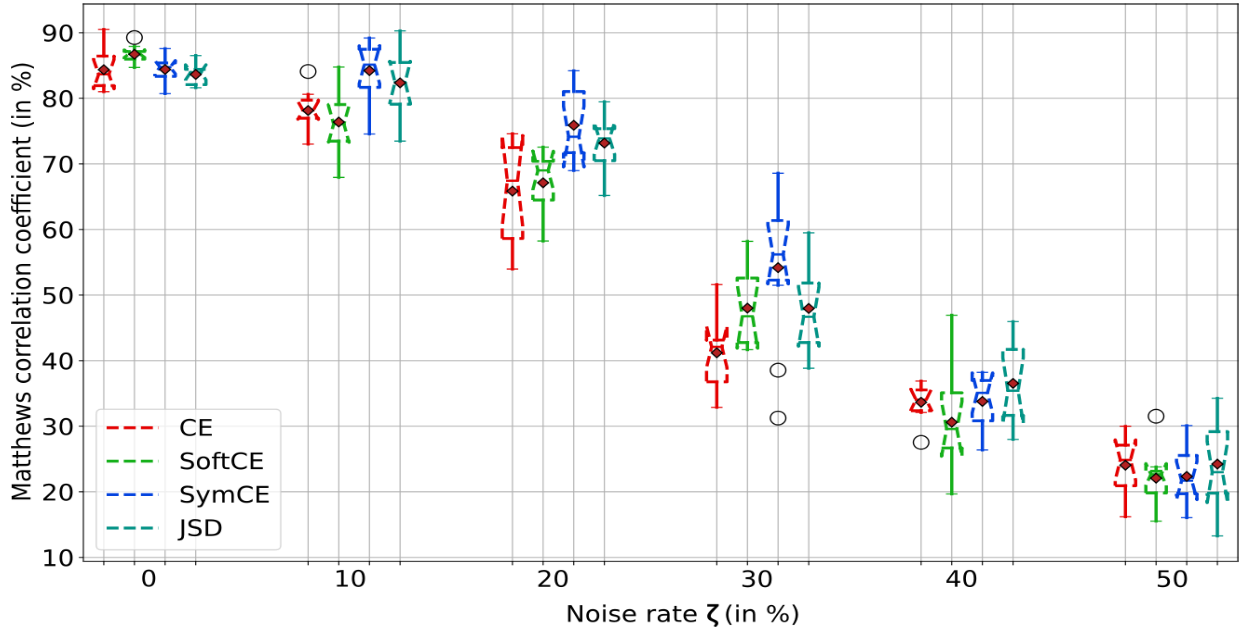


Figure - (a) HITS dataset without noise, **(b)** HITS dataset with 10% of symmetric noise in the soft labels.

Soft labels noise resistance



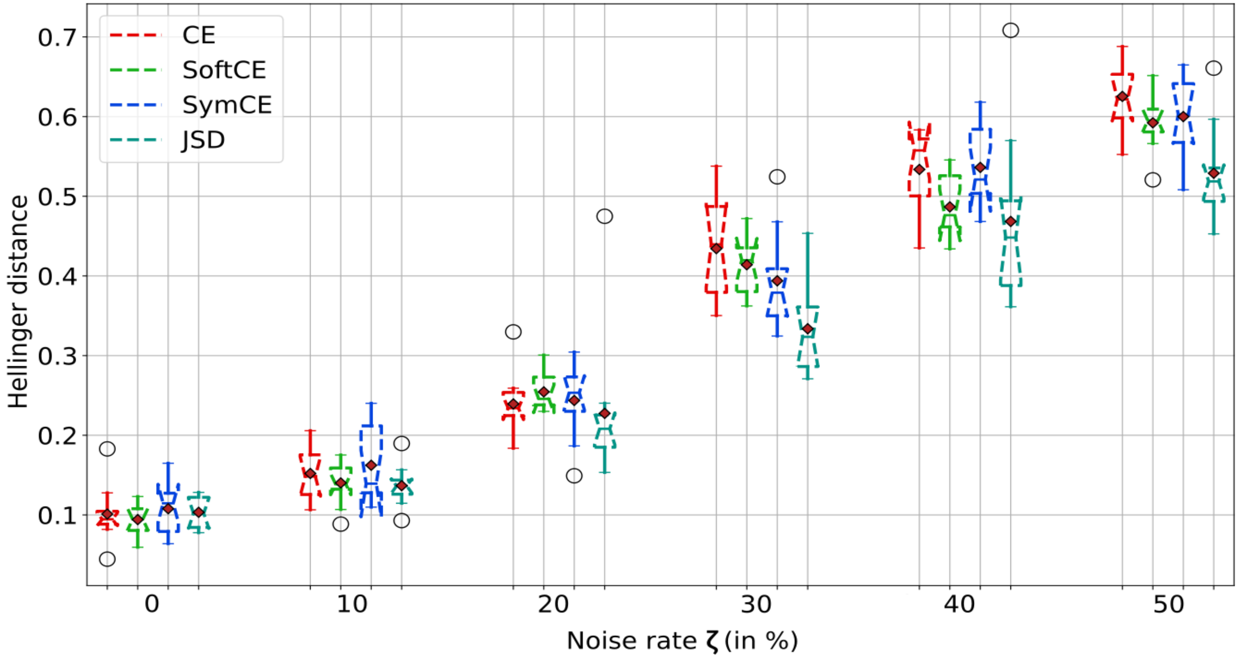
(a) 2D time-frequency CNN



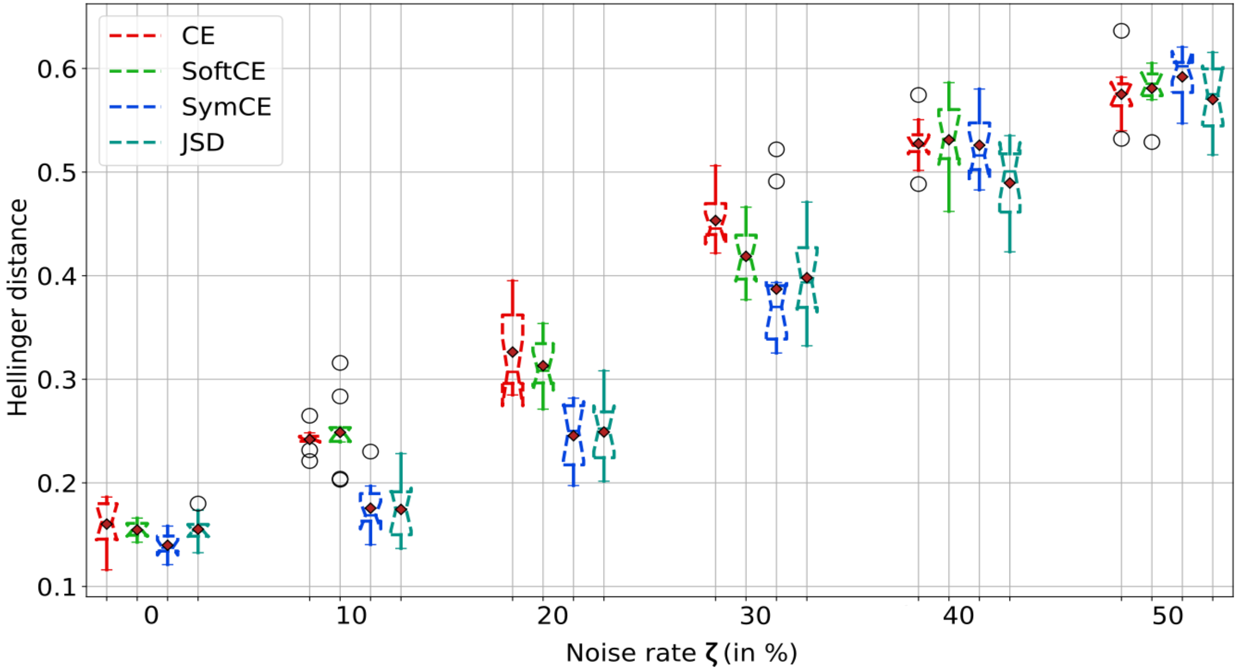
(b) Doppler signal 1D CNN-transformer

Figure - Classification performance of two types of models trained using the presented soft and hard label loss function

Uncertainty capturing



(a) 2D time-frequency CNN



(b) Doppler signal 1D CNN-transformer

Figure - Uncertainty capturing evaluation of two types of models trained using the presented soft and hard label loss function

Geometric Mean Jensen-Shannon Divergence (GEO JSD)

$$JSDR(P||Q) = \propto (1 - \propto)[\boldsymbol{\beta}(H(P,P) - H(Q,P)) + \underbrace{\boldsymbol{\Upsilon}(H(Q,Q) - H(P,Q))]}_{\text{Robustesse au bruit prouvée théoriquement}}$$

Résultats préliminaires GEO JSD

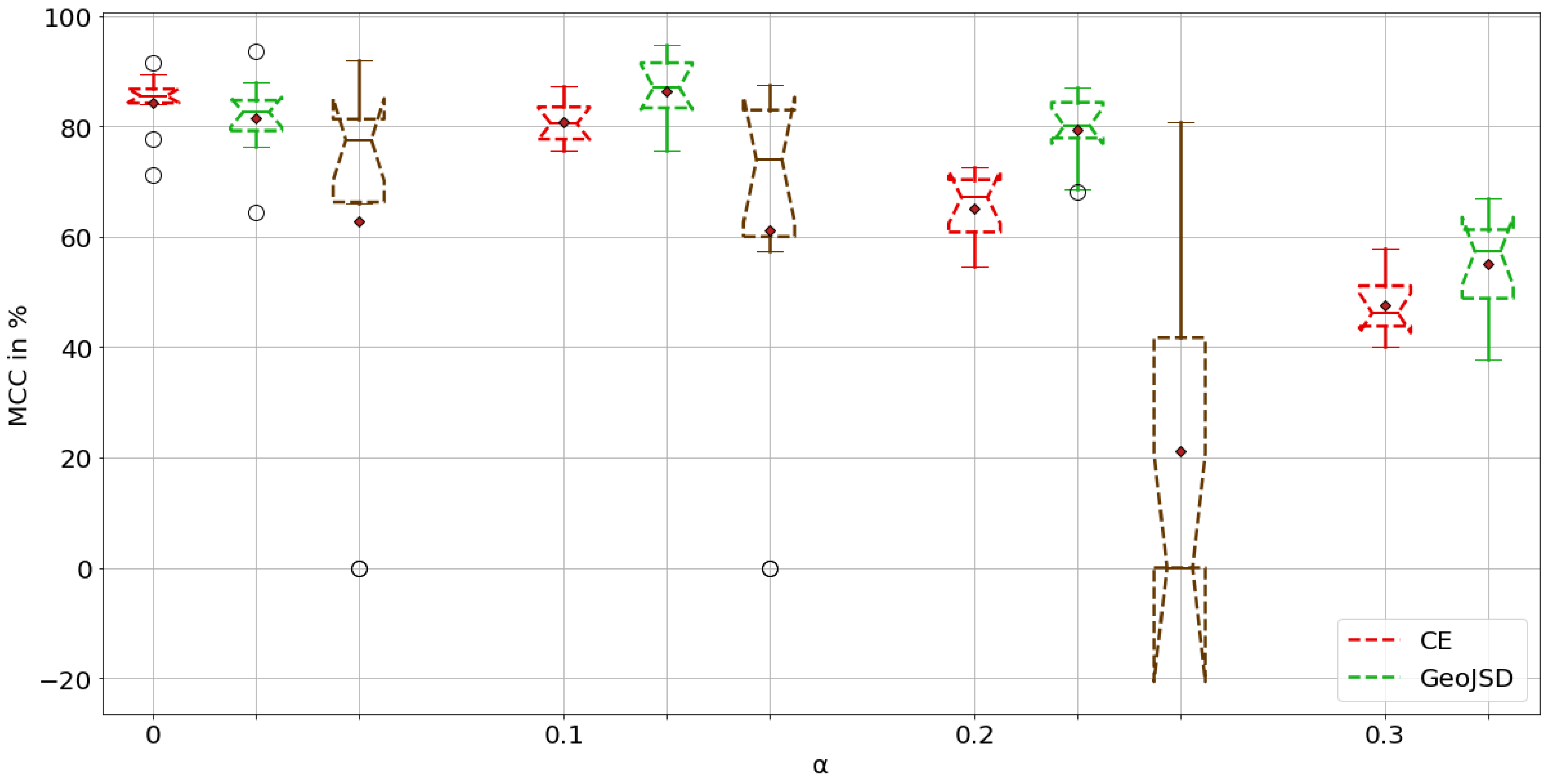


Figure – Résultats HITS-small

Résultats préliminaires GEO JSD

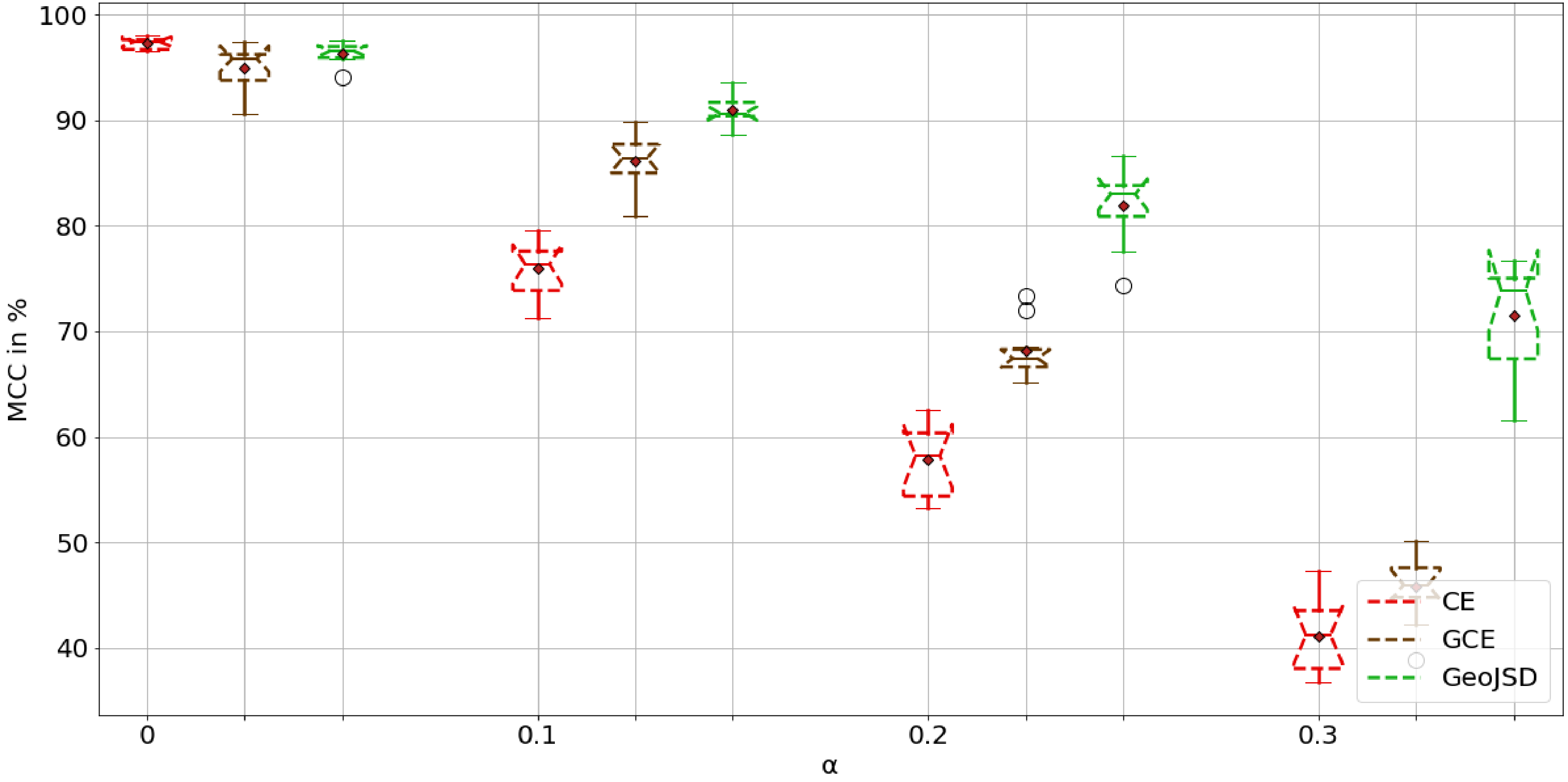


Figure – Résultats PTB

Mathews Correlation Coefficient (Binary classification)

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

$$\Rightarrow \quad MCC = 0 \Leftrightarrow TP \times TN = FP \times FN$$

Random classifier !

Statistical tests for comparison

- Important hypothesis for several statistical tests ➔ **Independence of observations**.
  - For k-fold cross-validation:
    - One sample belongs to the training dataset is k-1 times

The mean estimates are not independent !

  - For repeated holdout:
    - The training and testing datasets are fixed during repetitions.

- **Solutions**
  - Create different datasets, one per repetition ➔ Not yet possible for the HITS.
    - Reduces training and testing samples per dataset.
  - 5x2 cross-validation ➔ Difficult to do it subject-wise.
  - Other tests without independence hypothesis ?

Choice of q hyperparameter for GCE

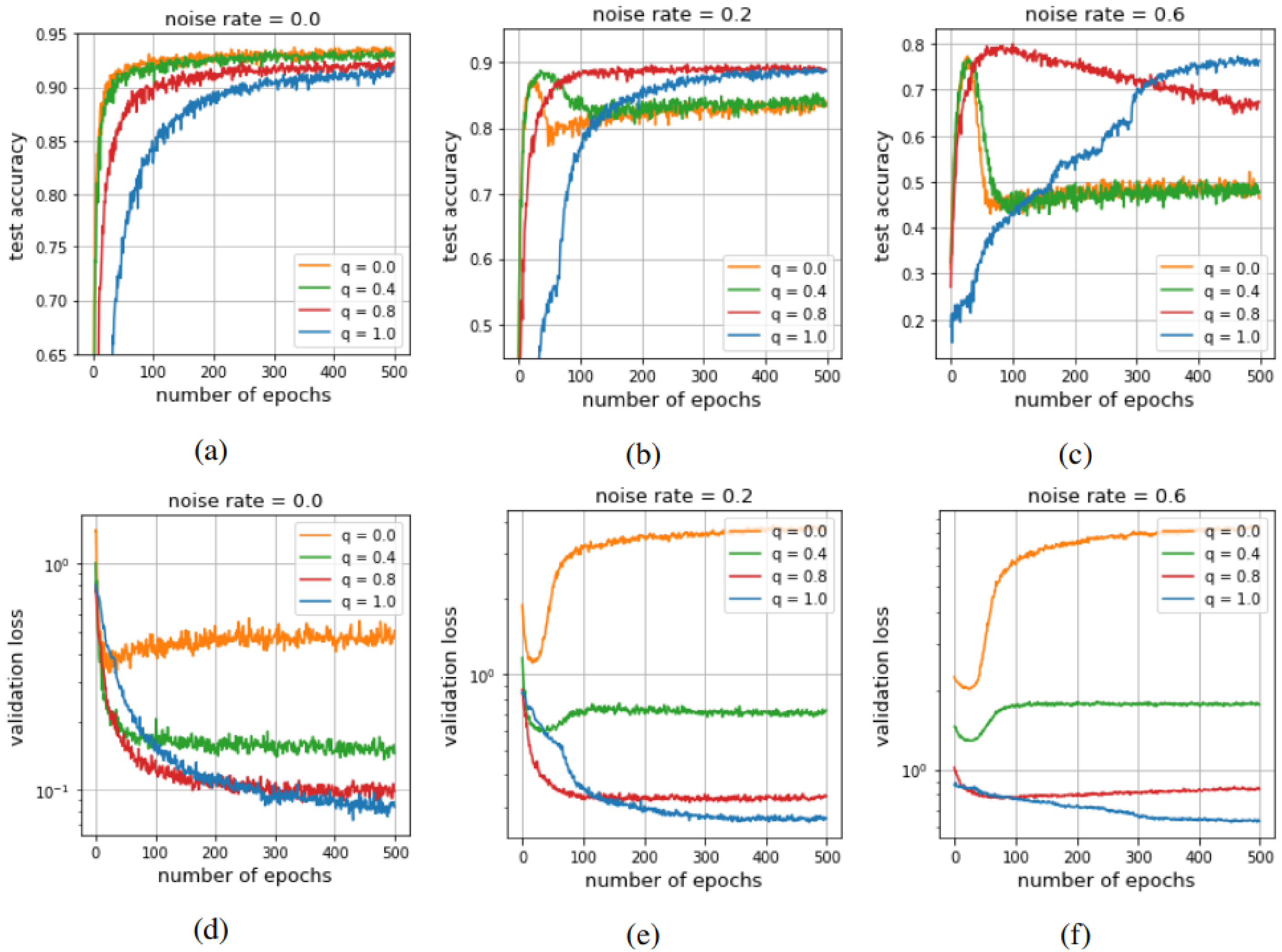


Figure – Test accuracy and validation GCE, for different values of q, on the XXX dataset (Zhang et Sabuncu 2018)

Automatic annotation vs Manual annotation

Split	No. patients	Total	SE	GE	Art.
Train	39	1541	456	610	6 198
Test	12	139	39	47	53

Table – HITS-small-I dataset.

Model	MCC
2D CNN	87.09 ± 4.31
1D CNN-trans.	79.17 ± 6.64
MIF-GR[illegible]	**91.89 ± 2.64**

Table – Multi-feature GDCE compared to single feature models on a noisy semi-automatically labeled dataset **HITS-small-I**.

Split	No. patients	Total	SE	GE	Art.
Train	40	7 264	456	610	6 198
Test	11	1 421	240	392	789

Table – HITS-sada dataset.

Model	MCC
2D CNN	84.03 ± 1.20
1D CNN-trans.	85.74 ± 1.16
MIF-GR[illegible]	**87.35 ± 0.85**

Table – Multi-feature GDCE compared to single feature models on a noisy semi-automatically labeled dataset **HITS-sada**.

Not the same test sets between experiments, so results are not directly comparable.

Test set of HITS-sada is harder.

Choice of the pruning parameters aTTQ

- **Choice of t_{min} and t_{max} is critical.**
  - **Bad choice:**
    - Poor classification performances.
    - Poor compression/energy performances
    - ➔ It **happens also for other methods** such as TTQ or TWN !
  - **Carefeul choice:**
    - **Great compression/energy/classification trade-off**.

- **Solutions**
  - Pruned trained ternary quantization with learnable threshold parameters !
  - Bayesian hyperparameter searching.
  - Larger study to understand influence of t_{min} and t_{max}.

Industrial application and impact on patient care

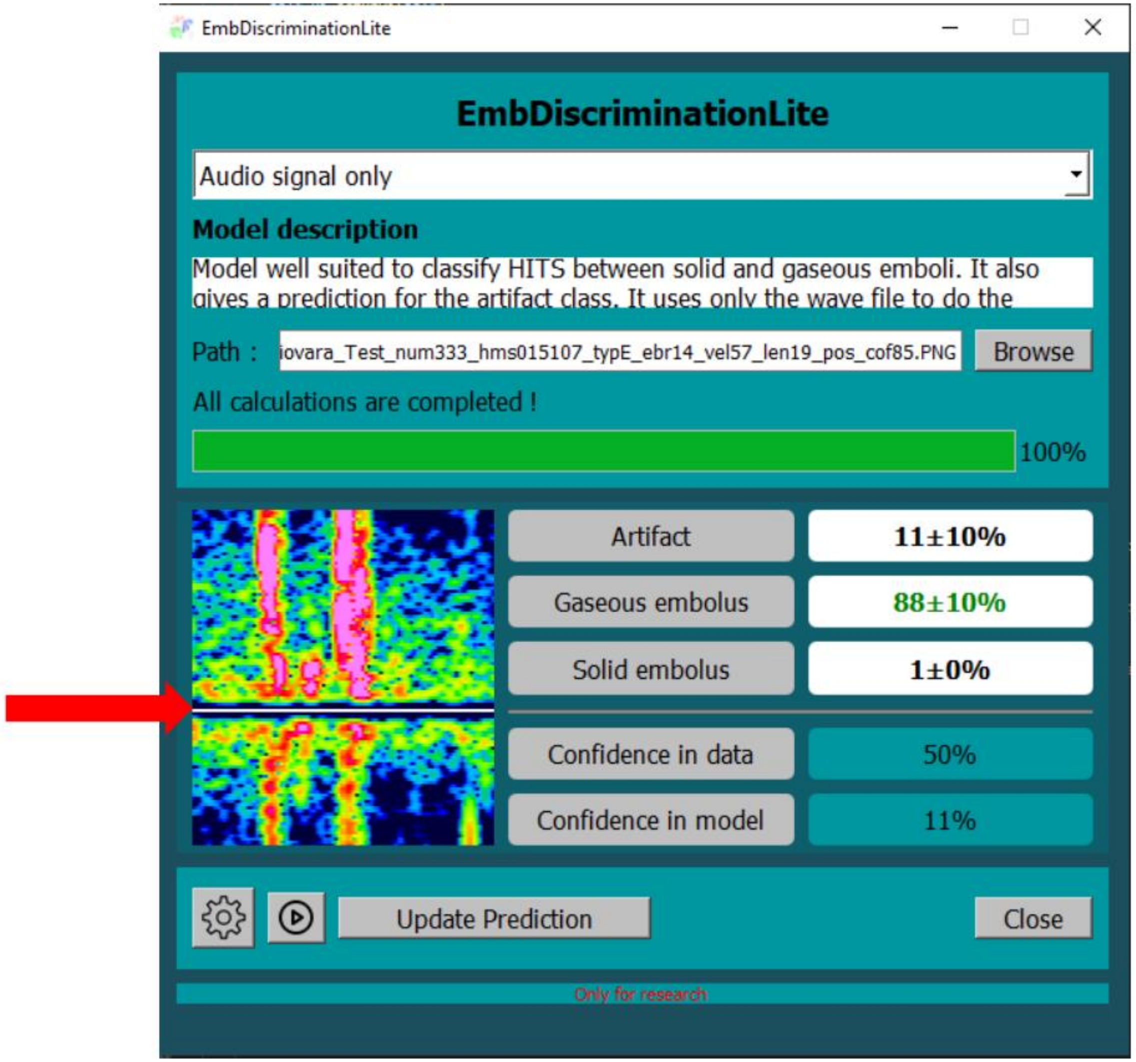
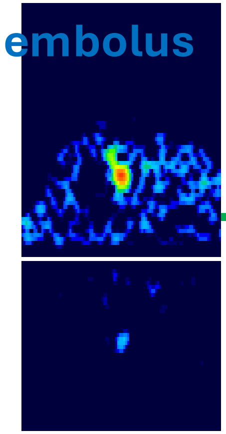


Figure – EmbDiscriminationLite application, deep learning classification module for ADMS for Atys Medical

Industrial application and impact on patient care

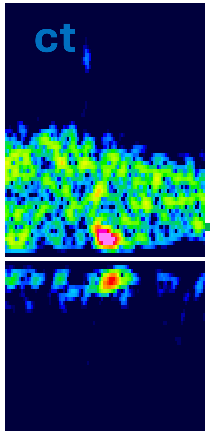
Solid embolus



Good diagnosis → **Adapted treatment ➔ Stroke prevention**

Bad diagnosis → **No treatment needed ➔ Stroke risk**

Artifact



Good diagnosis → **No treatment needed**

Bad diagnosis → **Wrong treatment ➔ Risk of complications**

Model compression comparison with other SOTA method

Dataset	Model	Quant. method	$CR_G^T \uparrow$	$CR_G^Q \uparrow$	$SRQW$	$EC_G^T \uparrow$	MCC $\uparrow$	Δ MCC $\uparrow$
HITS	2D CNN	FP	-	-	-	-	89.84 ± 3.09	-
		DoReFa	$\mathbf{89.18 \pm 0}$	$\mathbf{96.87 \pm 0}$	-	3.54 ± 0	85.05 ± 5.96	-4.79
		TTQ	24.96 ± 2.25	27.12 ± 2.44	28.96 ± 2.12	23.42 ± 1.30	$\mathit{86.82 \pm 2.29}$	$\mathit{-3.02}$
		aTTQ	42.98 ± 0.23	46.69 ± 0.25	$\mathit{45.95 \pm 0.21}$	$\mathit{44.04 \pm 0.19}$	86.14 ± 3.37	-3.70
		pTTQ	$\mathit{75.54 \pm 3.39}$	$\mathit{82.06 \pm 3.69}$	$\mathbf{83.12 \pm 3.47}$	$\mathbf{75.53 \pm 1.53}$	$\mathbf{89.33 \pm 4.45}$	$\mathbf{-0.55}$
	1D CNN-trans.	FP	-	-	-	-	82.64 ± 1.77	-
		DoReFa	$\mathbf{14.50 \pm 0}$	$\mathbf{96.87 \pm 0}$	-	0.37 ± 0.03	$\mathit{84.07 \pm 3.11}$	$\mathit{+1.43}$
		TTQ	0.14 ± 0.04	0.91 ± 0.27	6.75 ± 0.26	1.88 ± 0.03	83.22 ± 2.36	$+0.58$
		aTTQ	$\mathit{13.94 \pm 0.02}$	$\mathit{93.17 \pm 0.16}$	$\mathbf{93.53 \pm 0.15}$	$\mathbf{7.64 \pm 0.11}$	81.66 ± 4.17	-0.98
		pTTQ	8.37 ± 0.05	55.89 ± 0.34	$\mathit{58.50 \pm 0.32}$	$\mathit{2.01 \pm 0.05}$	$\mathbf{85.12 \pm 1.94}$	$\mathbf{+2.48}$
ESR	2D CNN	FP	-	-	-	-	92.81 ± 3.53	-
		DoReFa	$\mathbf{96.40 \pm 0}$	$\mathbf{96.87 \pm 0}$	-	29.90 ± 0	$\mathit{94.12 \pm 0.87}$	$\mathit{+1.31}$
		TTQ	85.61 ± 1.37	86.03 ± 1.37	86.59 ± 1.29	76.45 ± 1.13	$\mathbf{95.00 \pm 1.11}$	$\mathbf{+2.19}$
		aTTQ	88.48 ± 0.44	88.91 ± 0.45	$\mathit{89.30 \pm 0.42}$	$\mathit{84.49 \pm 0.33}$	92.41 ± 2.22	-0.40
		pTTQ	$\mathit{93.35 \pm 0.96}$	$\mathit{93.80 \pm 0.96}$	$\mathbf{94.17 \pm 0.91}$	$\mathbf{90.32 \pm 0.69}$	92.23 ± 2.32	-0.58
	1D CNN-trans.	FP	-	-	-	-	94.33 ± 1.51	-
		DoReFa	$\mathit{23.46 \pm 0}$	$\mathit{96.86 \pm 0}$	-	0.90 ± 0	$\mathbf{96.79 \pm 0.55}$	$\mathbf{+2.46}$
		TTQ	11.40 ± 2.61	47.07 ± 10.79	50.22 ± 10.16	3.21 ± 0.66	96.25 ± 0.79	$+1.92$
		aTTQ	21.02 ± 0.15	86.78 ± 0.63	$\mathit{87.59 \pm 0.59}$	$\mathit{5.37 \pm 0.04}$	95.34 ± 0.79	$+1.01$
		pTTQ	$\mathbf{23.86 \pm 0.04}$	$\mathbf{98.54 \pm 0.16}$	$\mathbf{98.67 \pm 0.15}$	$\mathbf{6.04 \pm 0.01}$	$\mathit{96.35 \pm 0.95}$	$\mathit{+2.02}$
MNIST	2D MNIST CNN	-	-	-	-	-	94.39 ± 0.46	-
		DoReFa	$\mathbf{51.67 \pm 0}$	$\mathbf{96.84 \pm 0}$	-	3.28 ± 0	87.03 ± 7.14	-7.36
		TTQ	13.86 ± 2.33	25.97 ± 4.37	30.40 ± 4.12	2.58 ± 0.35	$\mathbf{92.09 \pm 0.89}$	$\mathit{-2.30}$
		aTTQ	28.98 ± 1.26	54.32 ± 2.36	$\mathit{57.08 \pm 2.22}$	$\mathit{4.97 \pm 0.22}$	93.62 ± 0.96	$\mathbf{-0.77}$
		pTTQ	$\mathit{33.92 \pm 1.02}$	$\mathit{63.58 \pm 1.92}$	$\mathbf{65.79 \pm 1.80}$	$\mathbf{6.10 \pm 0.15}$	$\mathit{91.01 \pm 0.61}$	-3.38

Table – Comparison of aTTQ with other state-of-the-art methods

Difficulty of measuring energy consumption on real CPU/GPU

- **Energy consumption depends on**
  - Used hardware and the model.
  - Operations implementations.
  - Optimization of the trained models.

- **What is implemented ?**
  - Some sparse operations in PyTorch (Beta version).
  - Simulated quantization ➔ All operations are treated as 32 bits operations.

Solution ➔ **Simulation (difficult)**

- **Current codes do not allow efficient operations on common hardware**
  - Specialized hardware is needed for further improvements.