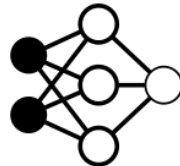


# Détection et Classification de signaux neuronaux

*Stage Expérimental*



## **I. Contextualisation**

1. Composante essentielle des neurosciences
2. Principes généraux
3. Objectifs du stage

## **II. Structure et Caractéristique du réseau**

1. Règles et inhibition
2. Structure des couches

## **III. Simulation de données biologiquement réalistes**

1. Mise en place des simulations
2. Outils d'analyse

## **IV. Création d'une interface graphique**

1. Choix des paramètres
2. Observations en temps réel

## **V. Étude sur la dépendance en bruit du Tri**

1. Résultats pour plusieurs SNR
2. Comparaison aux données d'origine

## **VI. Optimisation du réseau à haut bruit**

1. Travail sur la couche d'entrée
2. Propositions d'améliorations

## I. 1 Composante essentielle des neurosciences

- Rend compte de l'excitation neuronale
- Permet de remonter à une grande quantité d'information
- Ceci en temps réel

## I. 2 Principes généraux

- Utilisation d'un réseau de neurones pour la classification et la détection
- Spiking Neural Network (**SNN**)
- Mimétisme biologique plus économe

## I. 3 Objectifs du stage

- Génération de Signaux neuronaux par simulation de culture (**MEARec**)
- Mise en place d'une interface graphique pour suivre l'apprentissage du réseau
- Test de l'efficacité du réseau pour plusieurs rapport signal sur bruit (SNR)

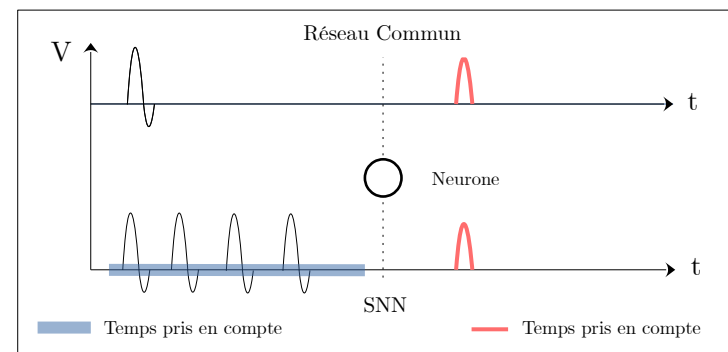


FIG 1. *Principe de fonctionnement d'un Spiking Neural Network (SNN)*

## II. 1 Règles et inhibition

- Règles de plasticité STDP et STP des synapses
- Renforcement de l'aspect bio-mimétique
- Inhibition par « *Winner Take All* » au sein d'une couche

## II. 2 Structure des couches

- Couche d'entrée : Neurons Sensoriels
  - $(1/\Delta VM + 2N_{overlap}, 10)$
  - Photographie
  - Totalemment connecté par STDP
- Couche intermédiaire : LIF
  - spécificité
  - Neurone d'attention
  - Totalemment connecté par STDP
- Couche de sortie : LIF / LTS
  - Encode les fragments
  - WTA

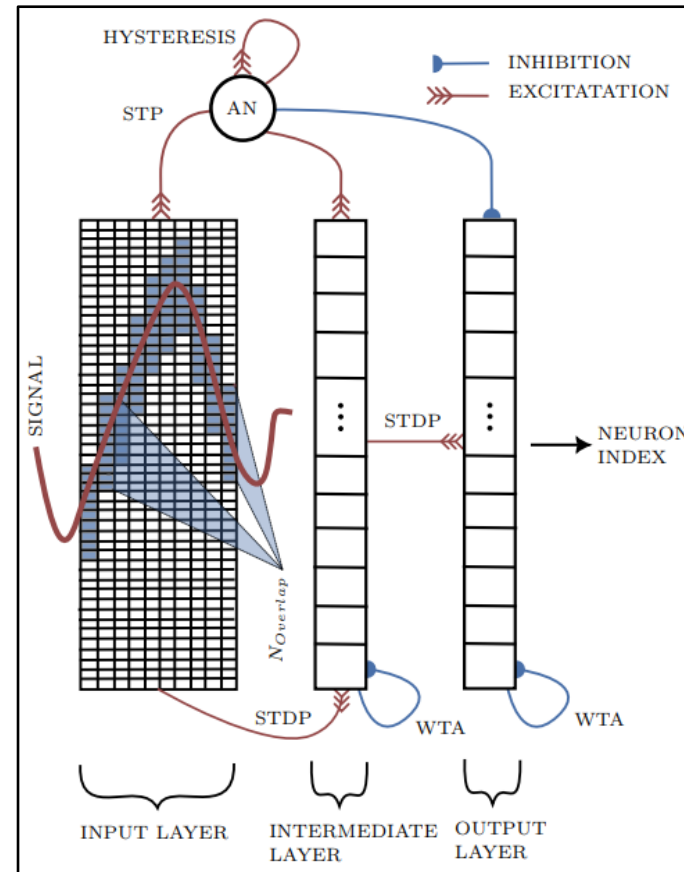


FIG 2. *Structure Globale du réseau de neurones*

### III. 1 Mise en place des simulations

- Objectif : Créer des données réalistes de neurones sur des électrodes (MEA) et faire varier le SNR
- Calibrage du SNR en fonction des différents paramètres des simulations :

- écart type du bruit, l'intervalle de variation de l'amplitude des potentiels d'action, le type de neurone, leur nombre, leur fréquence d'excitation et leur distance à l'électrode

- Il apparaît que l'écart type du bruit est le seul paramètre essentiel dans ceux étudiés, pour faire varier le SNR (FIG 3), en fixant l'amplitude des potentiels d'action.

- Calibrage du SNR (dépendance en bruit pour chaque expérience) :  
fit en a/x :  $SNR = \frac{\langle A_{max} \rangle}{\sigma_{noise}}$

- Choix des électrodes dans les simulations :

- Une seule électrode (une seule voie)
- Possibilités d'en générer plusieurs

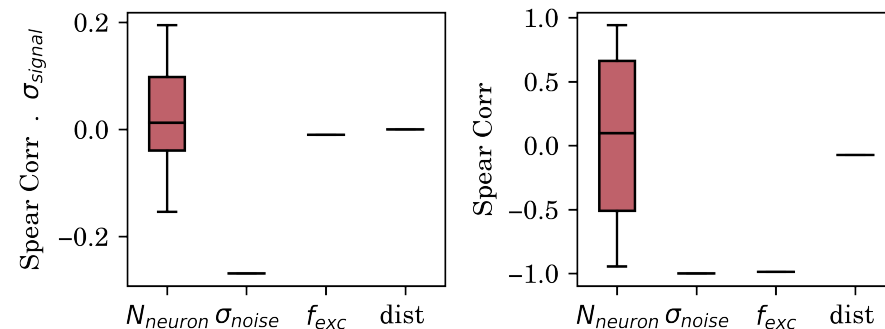


FIG 3. Corrélation entre paramètres et SNR

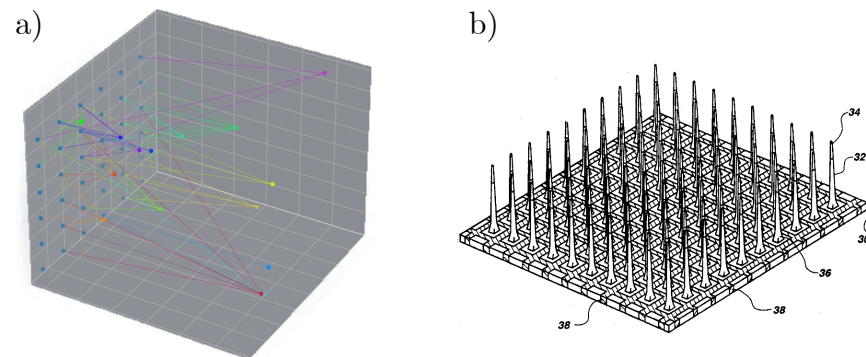


FIG 4. Schéma du système (a) et MEA (b)

## III. 2 Outils d'analyse des simulations

- Objectif : Créer des tous les outils nécessaires à la caractérisation des signaux pour chaque neurone simulé :
  - *Tables de vérité : localisation temporelle des potentiels d'action pour chaque neurone simulé (Sert à l'Évaluation du réseau de neurone) Voir FIG 5.c*
  - *Forme des potentiels d'action aux électrodes : Moyenne et Total Voir FIG 5.a*
  - *distribution temporelle des potentiels (ISI),*
  - *distribution de l'Intervalle de temps entre les groupes de potentiels ou bursts (IBI) Voir FIG 5.b*
  - *distribution de la durée des bursts.*

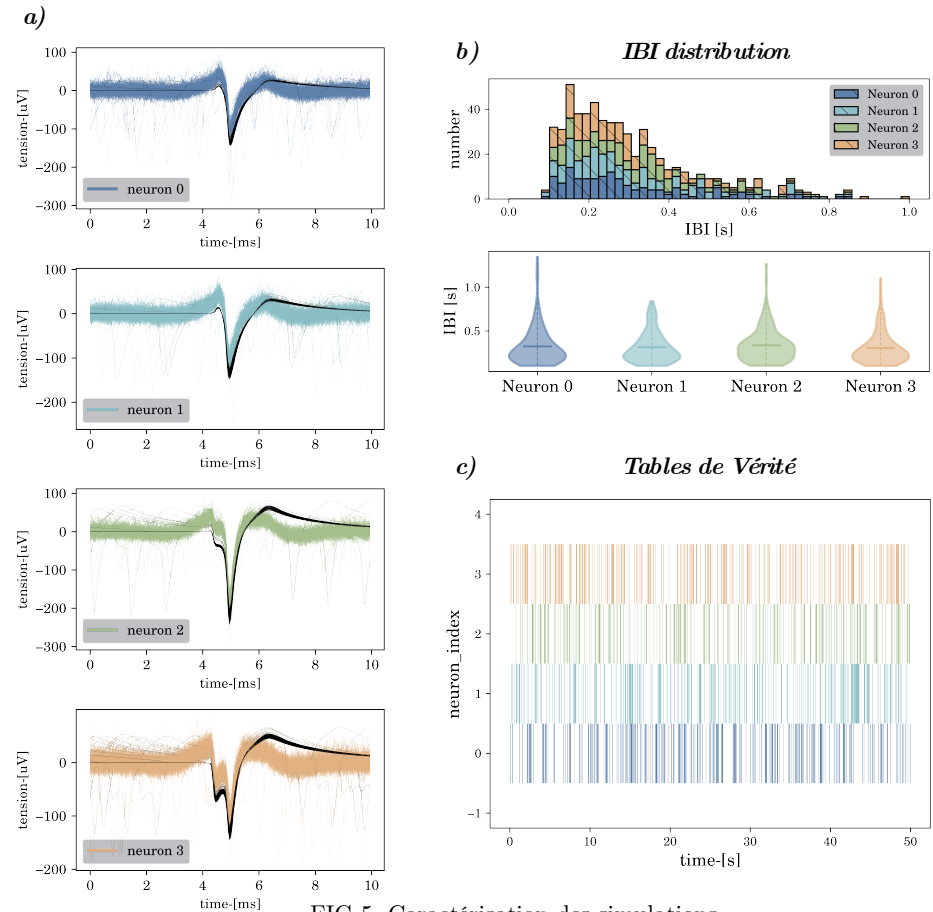


FIG 5. Caractérisation des simulations

## IV. 1 Choix des paramètres

- Création d'un onglet pour régler les paramètres des couches, des neurones les constituant à l'aide de **pyQT** et de **QtDesigner** (Fig 6), mise en parallèle de tous les processus avec des threads.

## IV. 2 Suivi en temps réel des performances du réseau

- Evolution du poids des synapses STDP pour chaque neurone de la couche intermédiaire.
- État d'excitation des différents neurones pour chaque couche
- Waveforms des potentiels détectés et leur attribution à chaque neurone de la couche de sortie
- Correspondance entre les poids des synapses STDP et le signal réel

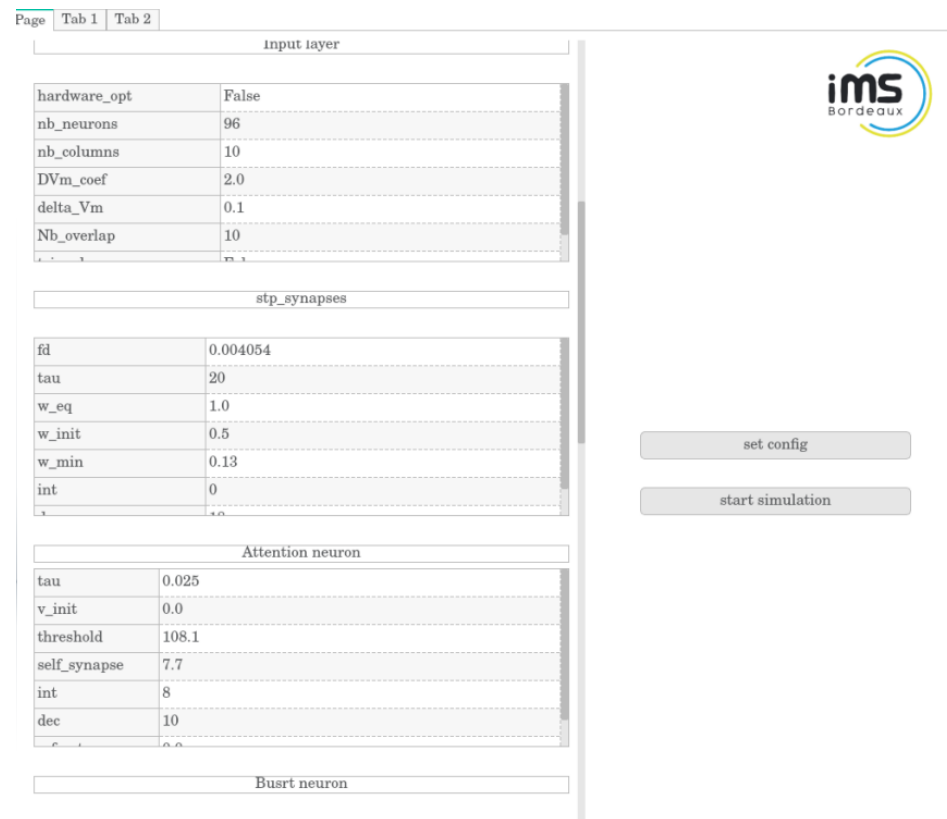


FIG 6. Interface de configuration

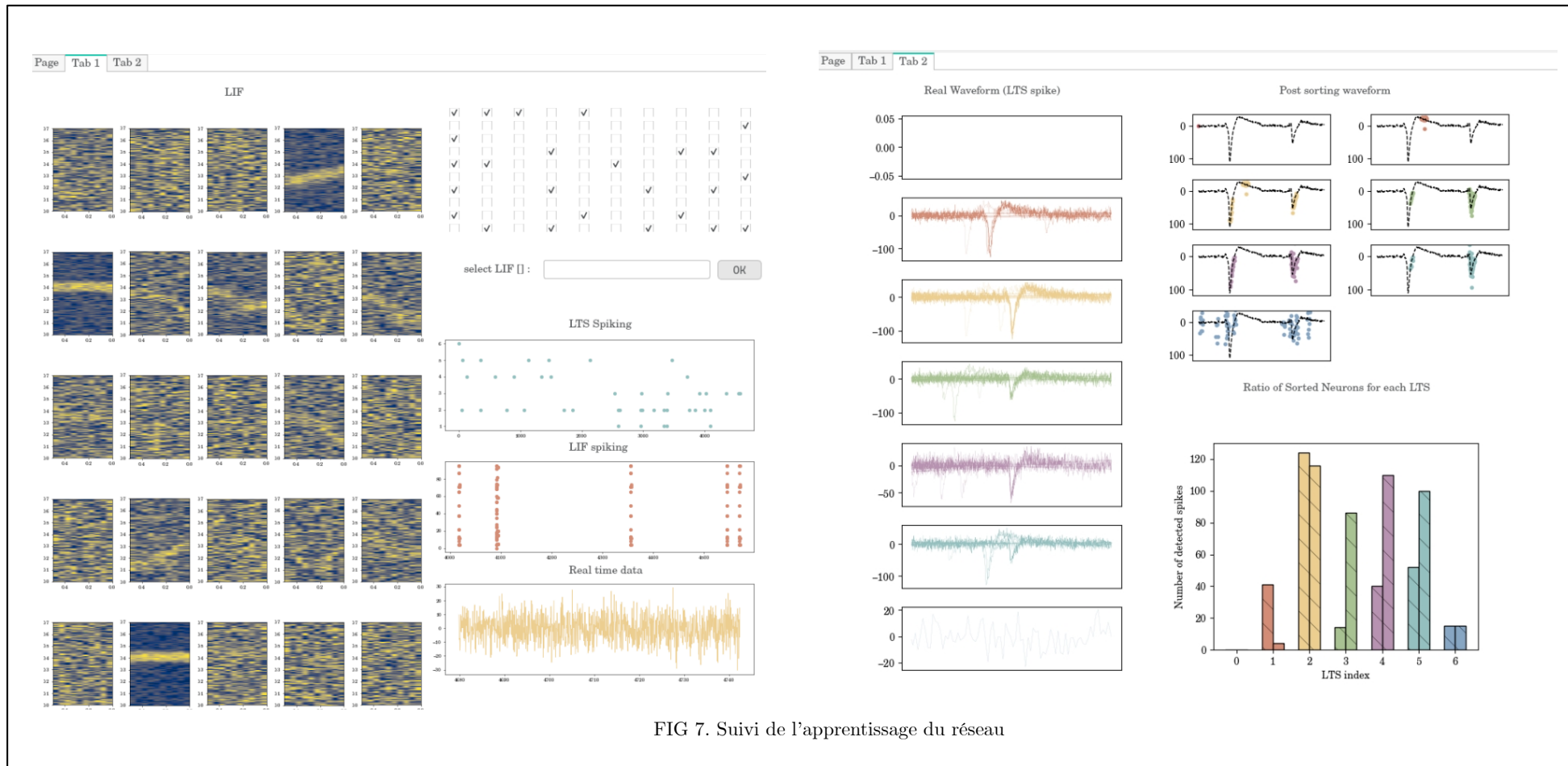


FIG 7. Suivi de l'apprentissage du réseau



## V. 1 Étude de la dépendance en bruit du réseau

- Au travers de 3 paramètres :
  - Precision (P)
  - Recall (R)
  - F Score (F)

$$\mathcal{P} = \frac{TP}{TP + FP}$$
$$\mathcal{R} = \frac{TP}{TP + FN}$$
$$\mathcal{F} = \frac{TP}{N_{Tot} + N_{Det}}$$

## V. 2 Comparaison aux résultats des anciennes données

- Le réseau présente de meilleurs résultats pour des SNR bas, alors que les anciennes données ne sont pas issues de simulations, ce qui est encourageant.
- Pour des SNR élevés les performances dans les deux premières couches sont moins bonnes (moins de neurones dans la couche d'entrée), compensées par couche de sortie.

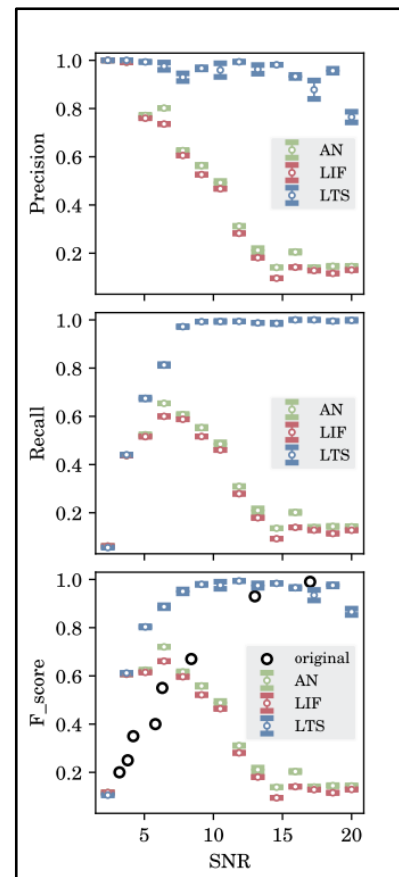


FIG 7. Dépendance en bruit des performances

## VI. 1 Optimisation de la couche d'entrée

- Impact de 2 paramètres qui conditionnent la qualité de l'échantillonnage fait par le réseau :
  - $N_{overlap}$  : Nombre de neurone s'activant en même temps dans une même colonne de la couche d'entrée.
  - $YvM$  : Coefficient de sensibilité de chaque neurone, conditionne le nombre de neurone sur une colonne.
- Il apparaît que à bas SNR :  $N_{overlap}$  de l'ordre de la dizaine est optimal pour avoir des métriques de l'ordre de 0.90, et ceci pour les deux couches qui apprennent.

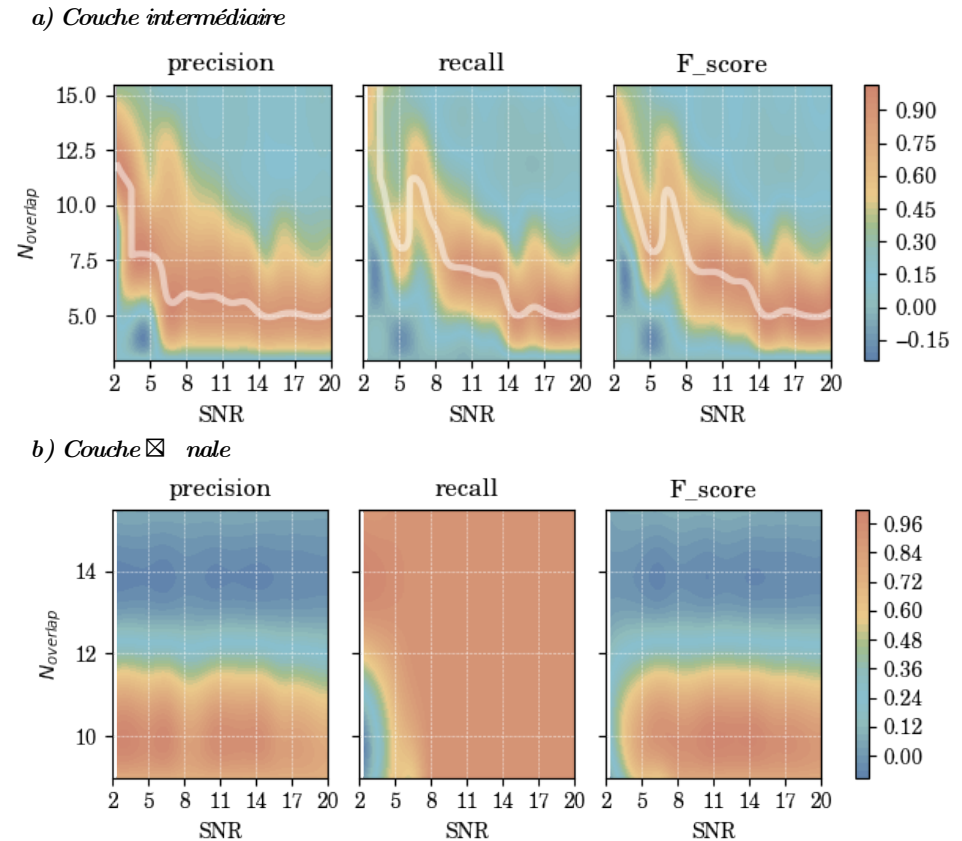


Fig 9. Effet de  $N_{overlap}$  et du SNR sur les performances

- Pour le coefficient de sensibilité  $YvM$ , l'étude s'est fait sur  $\alpha$ :

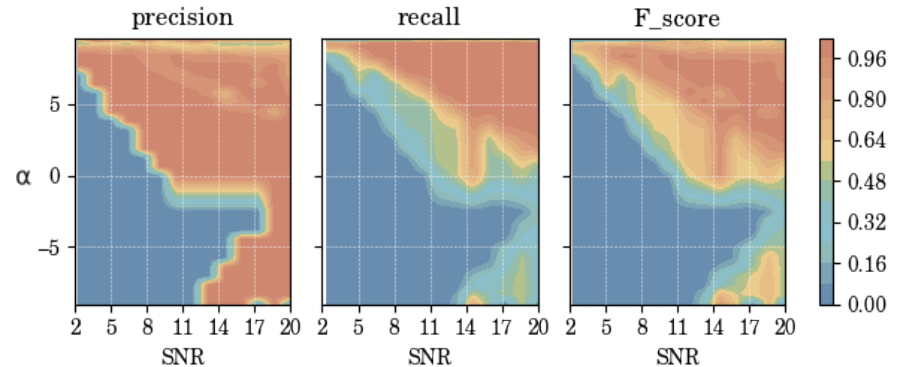
$$YvM = 10 - \alpha$$

A bas SNR : faible sensibilité (augmentation de la quantification du réseau), il nous faudra donc plus de ressources à bas SNR

## VI.2 Propositions d'améliorations

- Modifier la dépendance en bruit du nombre de neurones de la couche d'entrée (ici inversement proportionnel au bruit)
- Implémenter un réseau synaptique dans les simulations (augmenter la corrélation entre les neurones).

a) Couche intermédiaire



b) Couche  $\boxtimes$  nale

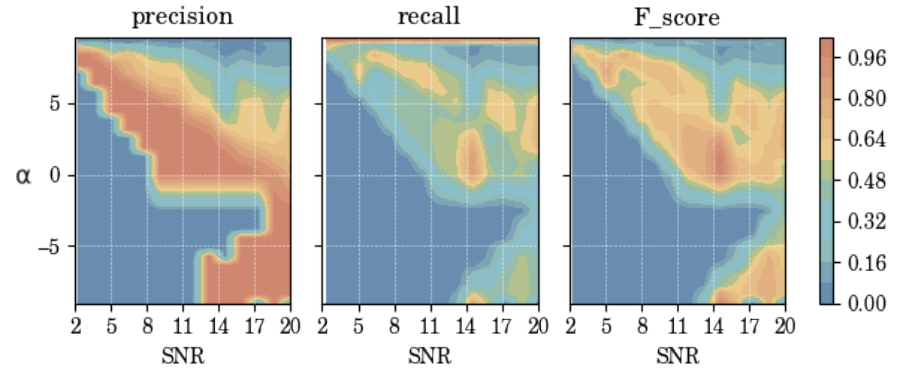


Fig 10.  $\boxtimes$  et  $dYvM$  sur les performances

## Conclusion

- 3 axes différents :
  - Génération de signaux
  - Mise en parallèle du code et implémentation d'une interface
  - Optimisation de la couche sensoriel
- Outils très souple : tout signaux numériques stochastiques
- Application en physique pour classifier des perturbations ...
- Application dans le domaine médicale, où l'apprentissage doit être adapté à chaque patient (mise en place de prothèses)

Merci de votre attention