

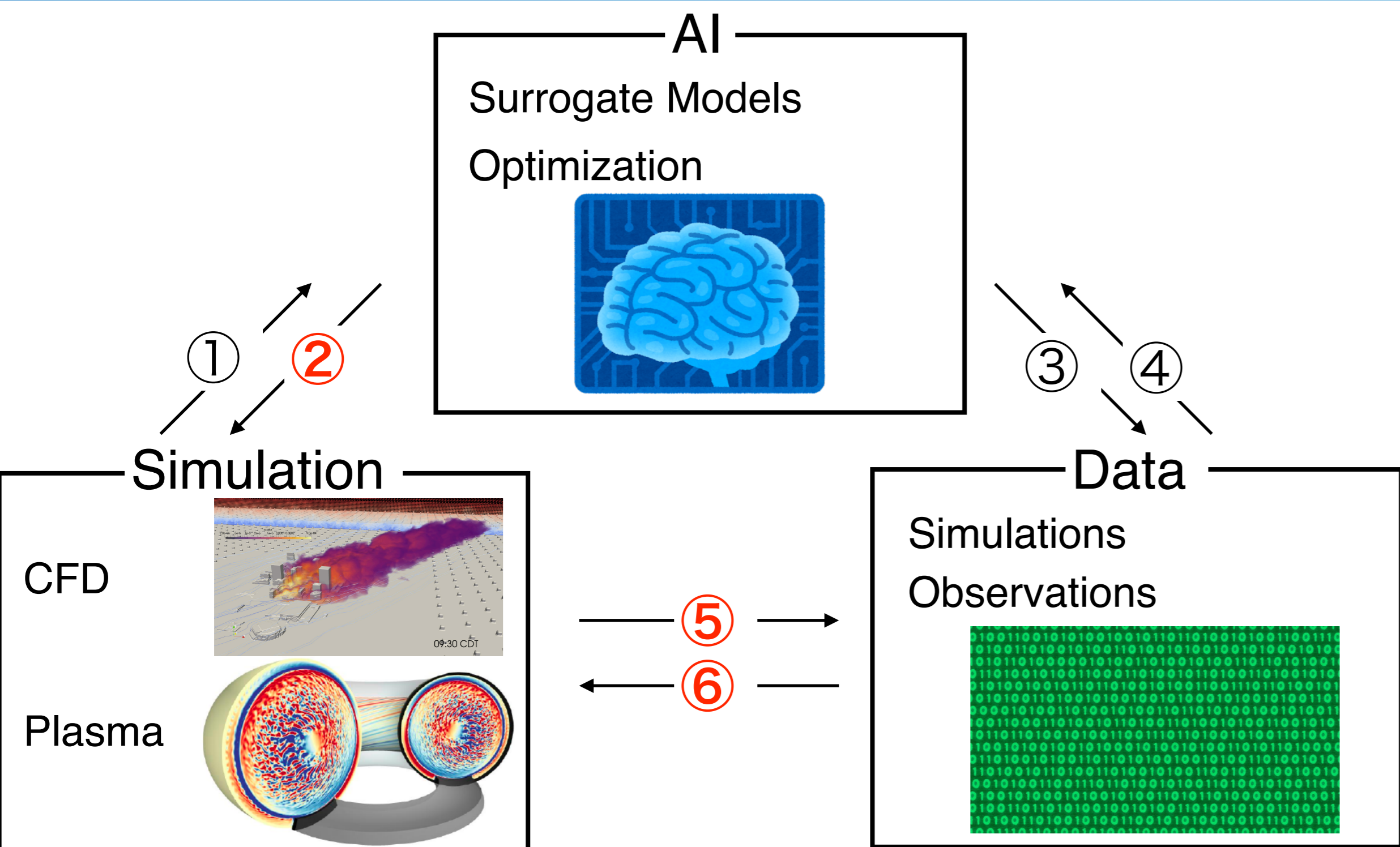


jh230033 (国際課題)

# Developing AI-assisted high performance fluid simulation codes

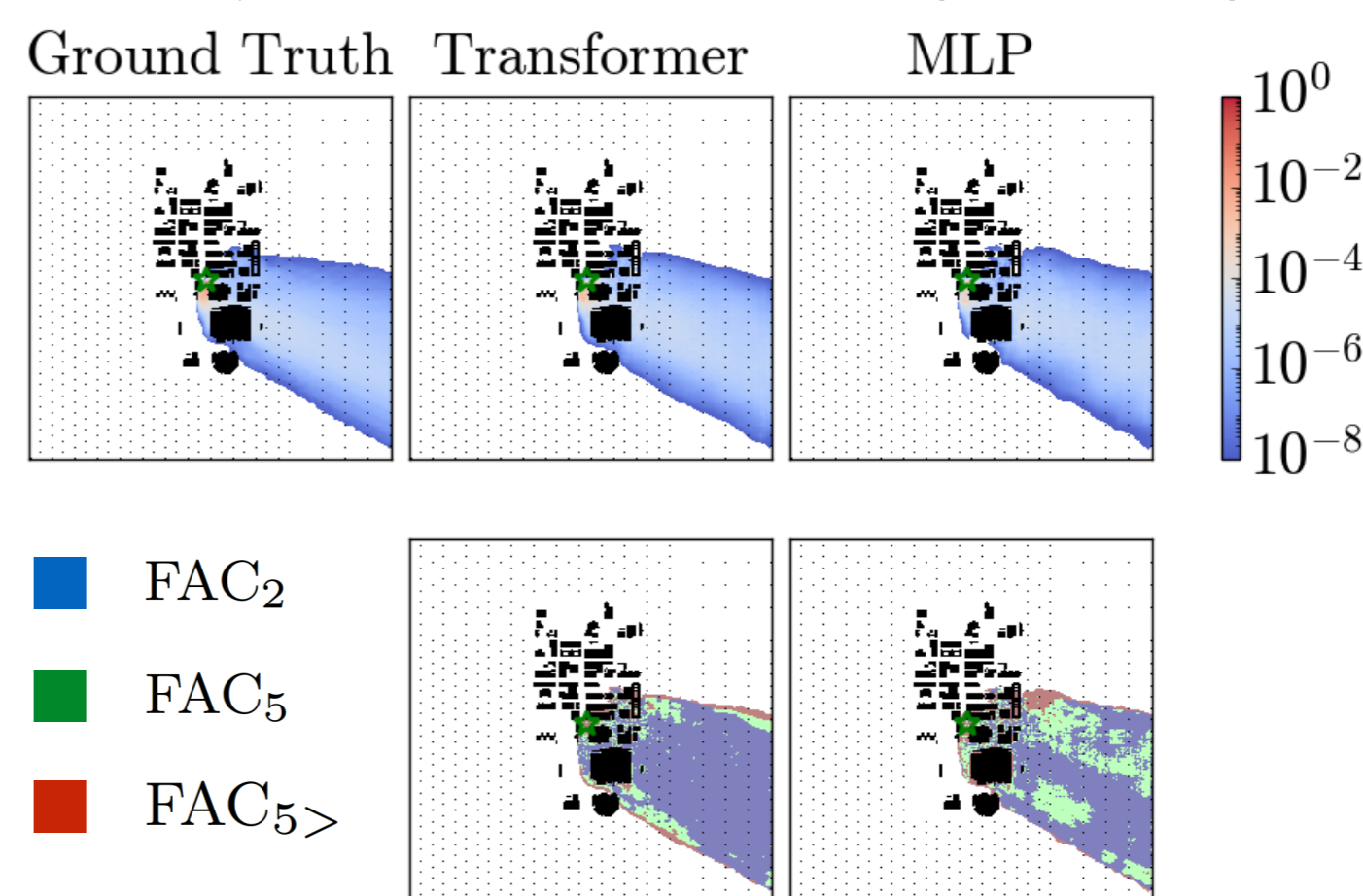
Y. Asahi<sup>1</sup>, J. Bigot<sup>2</sup>, S. Maeyama<sup>3</sup>, X. Garbet<sup>4</sup>, V. Grandgirard<sup>4</sup>, K. Obrejan<sup>4</sup>, T. Padioleau<sup>2</sup>,  
T. Shimokawabe<sup>5</sup>, H. Shiba<sup>6</sup>, T-H. Watanabe<sup>3</sup>, Y. Idomura<sup>1</sup>, T. Aoki<sup>7</sup>, K. Fujii<sup>8</sup>, N. Onodera<sup>1</sup>, Y. Hasegawa<sup>1</sup>  
1. JAEA, 2. Maison de la Simulation, 3. Nagoya Univ, 4. CEA, 5. The University of Tokyo,  
6. University of Hyogo, 7. Tokyo Tech., 8. ORNL

## 研究計画全体の概要



- ①: On going simulation data for in-situ fine tuning
  - ②: **AI based steering of simulations**
  - ③: Data compression, sanity check for simulation data [3]
  - ④: Surrogate models [1, 2]
  - ⑤: **Create simulation dataset needed for the AI model**
  - ⑥: **Data assimilation**
- 赤字: jh230033の内容

都市区画汚染物質拡散の即時予測モデル [1]  
時系列観測データと放出点位置から拡散の即時予測



$$FAC_x = \frac{1}{n} \sum_{i=1}^n N_i, \text{ with } N_i = \begin{cases} 1 & \text{for } \frac{1}{x} \leq \frac{C_{p,i}}{C_{r,i}} \leq x, \\ 0 & \text{else,} \end{cases}$$

[1] Y. Asahi, et al., Boundary Layer Meteorology (2023)  
[2] Y. Asahi, et al, AI4S, IEEE CLUSTER 2021  
[3] Y. Asahi, et al, Phys. Plasmas (2021)

HPC: **senders/receivers**を利用した非同期かつ性能可搬な実装手法

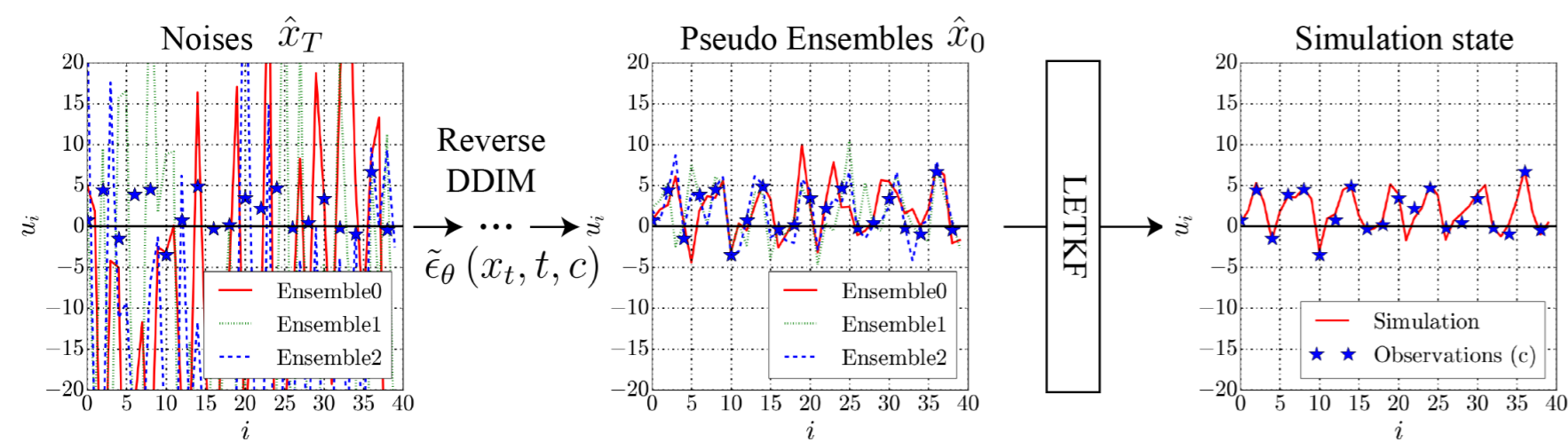
データ同化など複雑なワークフローの非同期実行

AI: ニューラルネットワークを用いた**データ同化** (計算と観測のずれを抑える) モデルや**サブグリッドスケール** (SGS)モデルの開発

## AIによるデータ同化モデルの開発 (Lorenz96モデルへの適用事例)

### NNデータ同化モデルの開発 [4]

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F (\forall i = 1, \dots, N)$$

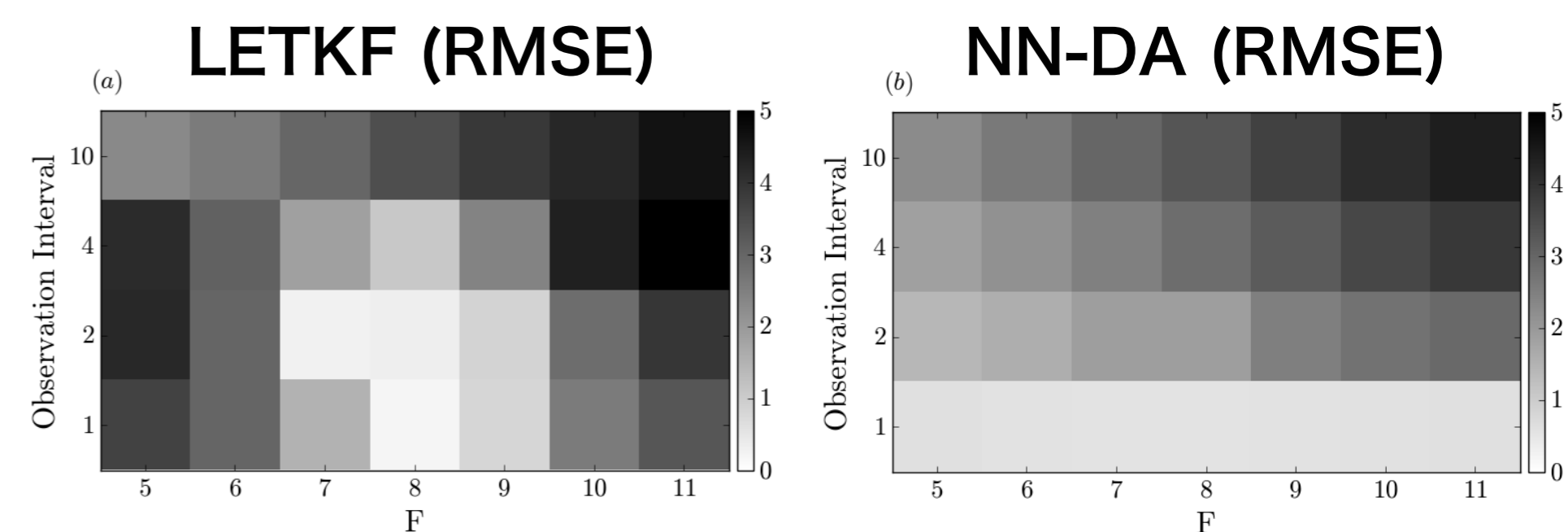
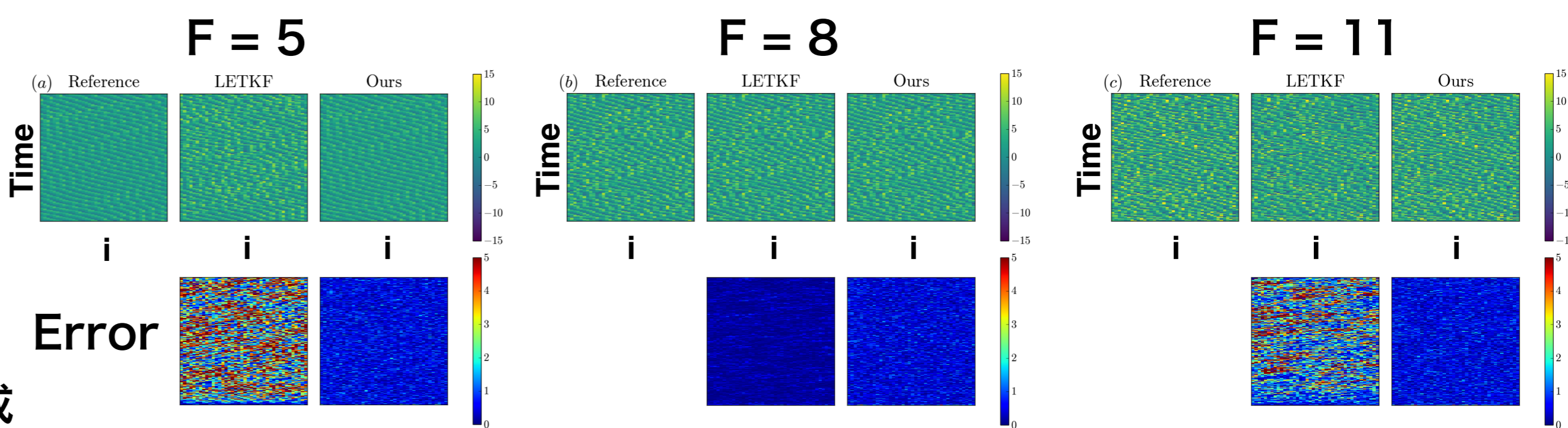


観測 (c)をガイダンスとしたDiffusionモデルの作成

Diffusion modelによって**疎かつノイズがある観測データから擬似アンサンブル**を生成し、これを用いてLocal ensemble transform Kalman Filter (LETKF)で同化する

1. LETKFで問題なモデルエラー (Fが異なる場合)に対してロバスト (アンサンブルの分散が常に保持され、filter divergenceを回避)
2. 高コストなアンサンブル計算不要

シミュレーション (F=8で固定)と観測 (Fが変化)でFが異なる場合のデータ同化例 (観測間引きなし)



[4] Y. Asahi, et al., SynS&ML @ ICML2023

## C++26のsenders/receiversを用いた非同期かつ性能可搬なHPC+AI流体アプリケーションの実現

### 3D熱方程式における通信演算オーバーラップ

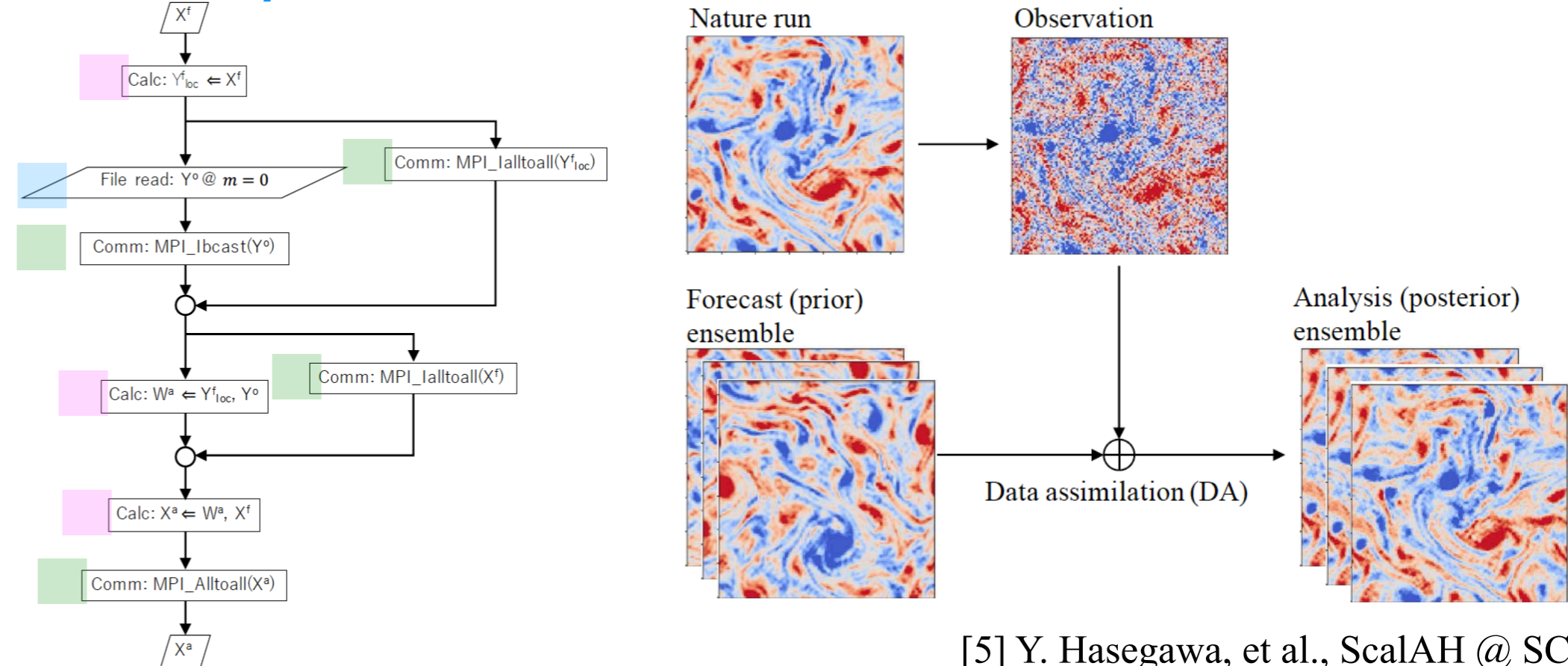
```
for(std::size_t i=0; i<conf.nbiter_; i++) {
  comm.pack(scheduler, u);
  auto inner_update = stdexec::when_all(
    stdexec::just() | exec::on( scheduler,
      stdexec::bulk(n, heat3d_funcor(conf, u, un) ) ),
    stdexec::just() | stdexec::then( [&]{ comm.commP2P(); } )
  );
  stdexec::sync_wait( std::move(inner_update) );
  comm.boundaryUpdate(conf, scheduler, un);
  std::swap(u, un);
}
```

**Overlapped**

senders/receivers [https://github.com/NVIDIA/stdexec]によるGPUカーネル (熱方程式ソルバー) とMPI通信 (P2P)の非同期実行

### Performance PortableなLETKF [5] 実装

演算、通信、I/Oが絡むデータ同化のワークフローの最適化



[5] Y. Hasegawa, et al., ScalAH @ SC22