2024年VRChat自律機械知能プロジェクト
中間報告

げそん＜GesonAnko＞
Myxy
Zassou
ぶんちん
田中スイセン
Earl Klutz（クルツ）

最初のお願い

- 本LTは視聴している皆さんを**Critic(直訳:映画の評論家)**と想定して作られています。

- **面白そうな点**や**よくわからない点**、**突っ込みどころ**など是非考えながら視聴してみてください！
  そして後で**ツッコミ入れて**ください！

最初のお願い

- 本LTは視聴している皆さんを**Critic(直訳:映画の評論家)**と想定して作られています

難しい質問はげそんさんに丸投げするよ！
やったね！

VRChat自律機械知能プロジェクトとは

- VRChat上に**自律機械知能を生み出し**、
  その振る舞いを記録・解析することにより
  人間及び知能全体への理解を深める。

- また、それらを論文にまとめバ学会にて発表する。

PAMI<Q>好奇心ベースの強化学習（予測誤差ベース）

1. 次にを起こることを予測

   Forward Dynamicsモデル f

   f :状態、行動→次の状態

   過去の経験から学習、予測誤差を最小化

2. 予測誤差（驚き）を報酬化

   仮定：未学習なことは予測誤差が大きい

3. 報酬を最大化する行動を生成

   Policy モデル　π

   π ：状態→行動

   強化学習の枠組みで学習

PAMI<Q>好奇心ベースの強化学習（予測誤差ベース）

1. 次にを起

Forwa

f :状態

過去の経験

2. 予 き） を報酬化

ことは予測誤差が大きい

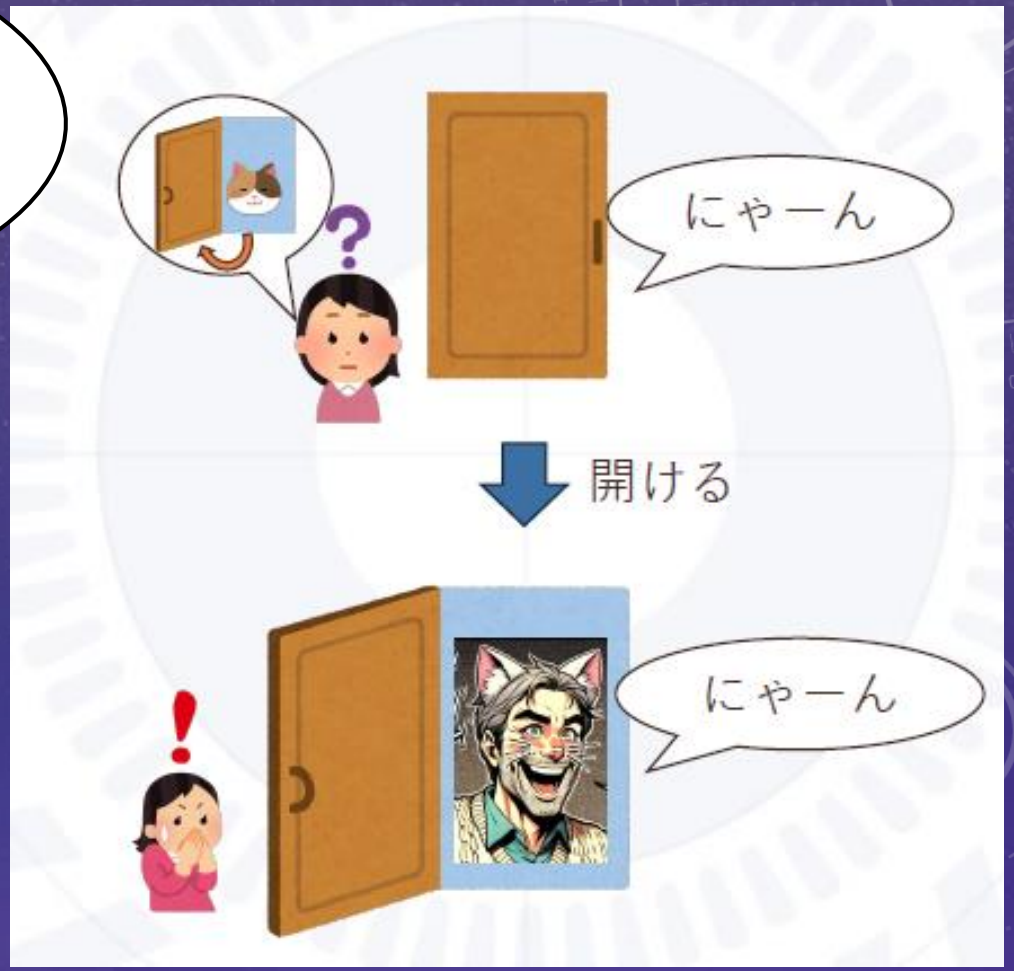
3. 化する行動を生成

π

動

組みで学習

予想したものと実際のものが
違えば違うほど高い報酬になる
というのがポイントだよ！



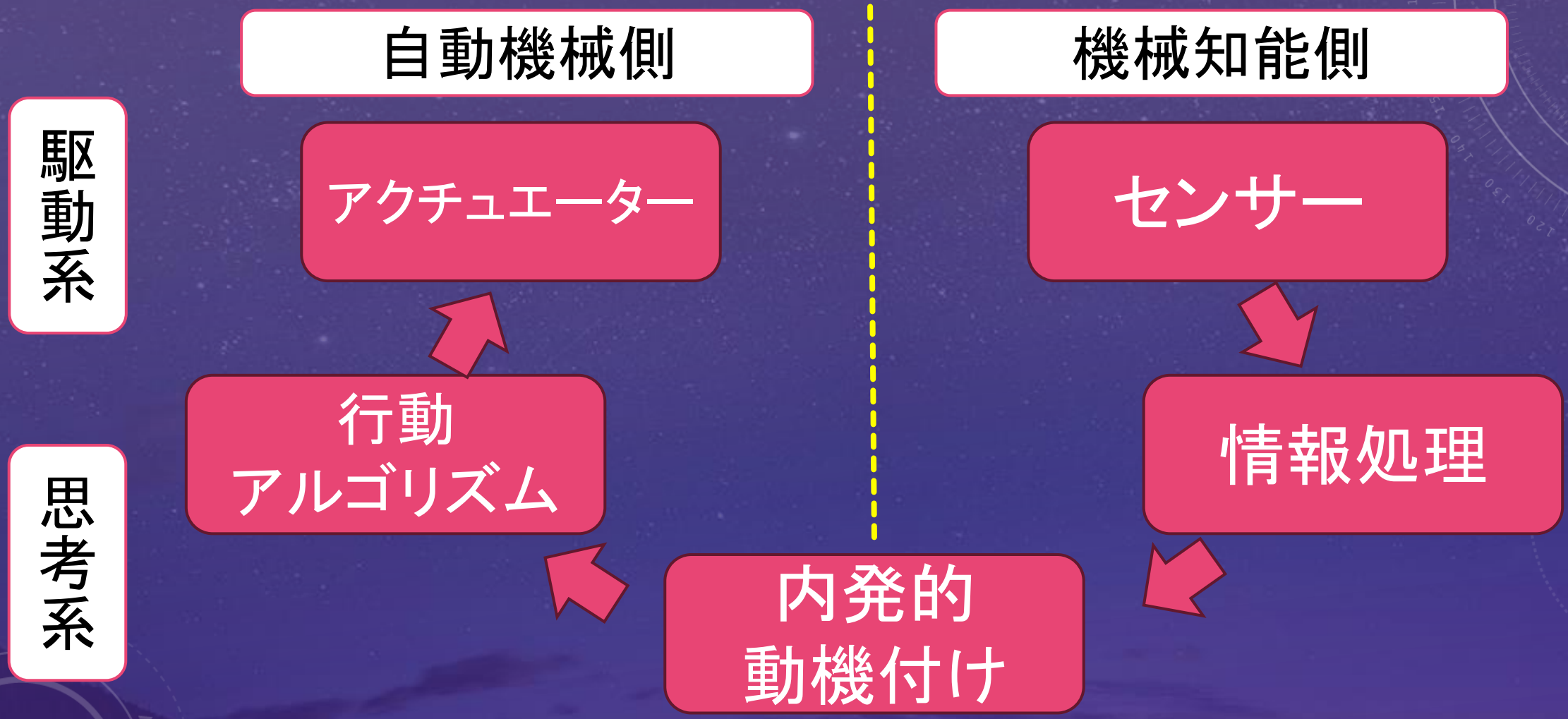
昨年の成果物

昨年の原始自律機械知能 P-AMI<Q> 実装

- Primitive Autonomous Machine Intelligence on Q(Cu)riosity.
- 原始的ではあるが、**好奇心ベース**の
  内発的動機付けを報酬に学習し
  探査し続けることが出来るP-AMI<Q>

- 実装担当は以下の3名。
  げそんさん、ocha_krgさん、myxyさん

VRChat上で動作する自律機械知能　基本構造



VRChat上で動作する自律機械知能　基本構造

自動機械側

駆動系

アクチュエーター

行動
アルゴリズム

思考系

内発的
動機付け

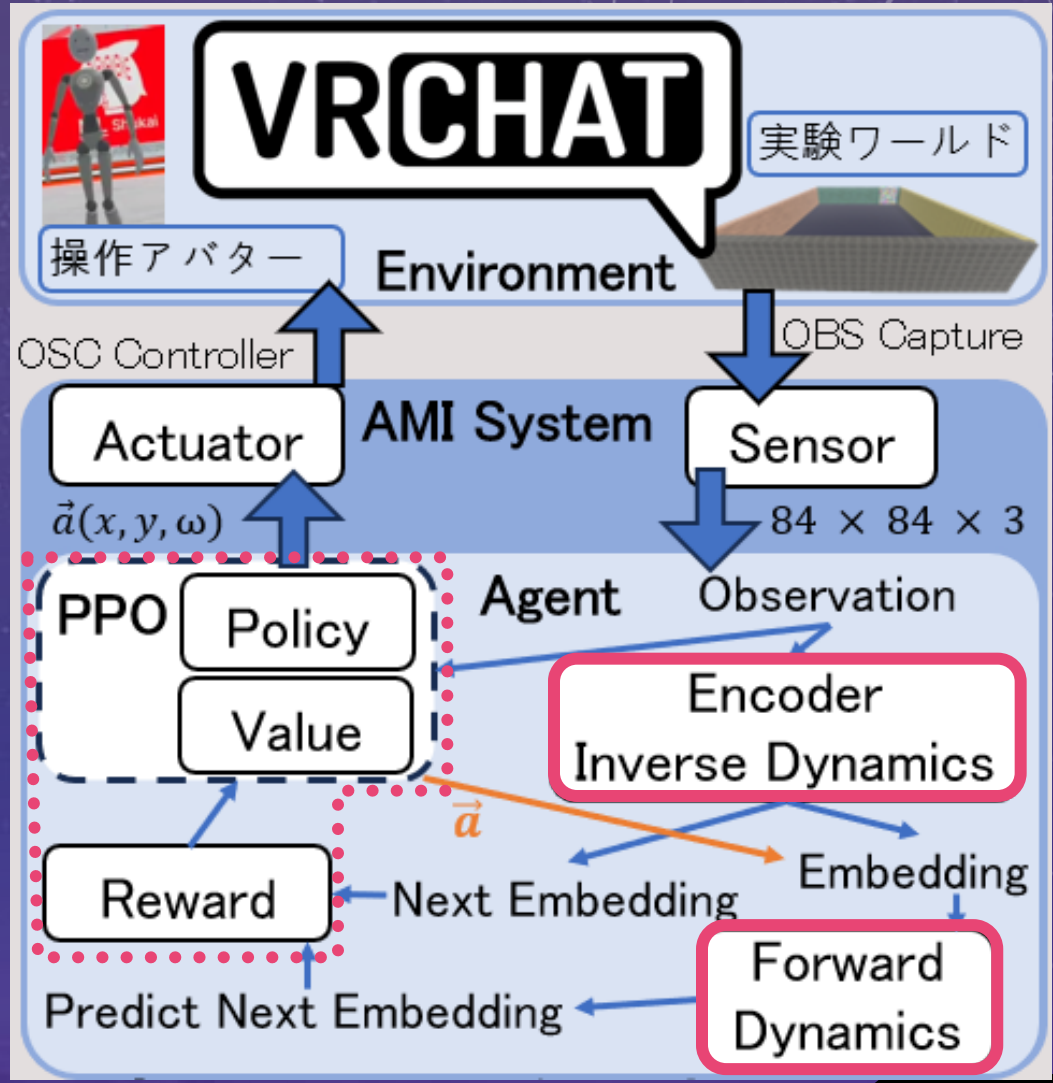
情報処

センサー

処理は複雑になってるけど
前も今も基本的な考え方は
同じだよ！

去年のP-AMI<Q>

- 基本構造は内発的好奇心を報酬とした学習構造
  - 観測Encoderで**特徴量抽出**
  - Forward Dynamicsで**行動後を予測**
  - 特徴量と予測から**好奇心の報酬**を算出
  - 強化学習で次の**行動を決定**



課題点

- 非同期で行動していないため**定期的に止まる**
- 最低限の原始的な状態なため未学習状態から学習開始
- モデルサイズを大きくできない
- 計算リソースの非効率的使用

2024年度版　P-AMI<Q>　大きな変更点

1.　処理手続きを3つのスレッドにし非同期化

- Mainスレッド、Inferenceスレッド、Trainingスレッド

2.　推論と学習のそれぞれのデータとモデルの効率的な処理

- Inferenceスレッドで収集と推論を行いTraningスレッドにデータを渡し結合
- 学習したモデルを素早く切り替えてからTraningスレッドでモデルをコピー

3.　各学習モデルは新方式にそれぞれ変更

- 観測エンコーダ：I-JEPA
- Forward Dynamics：SioConv
- 強化学習方式：~~Dreamer~~　→去年と同じくPPO

2024年度版　P-AMI<Q>　大きな変更点

1. 処理手続きを3つのスレッドにし非同期化
   - Mainスレッド、Inferenceスレッド、Trainingスレッド

2. **推論と学習**のそれぞれの**データとモデル**の効率的な処理
   - Inferenceスレッドで収集と推論を行いTraningスレッドにデータを渡し結合
   - 学習したモデルを素早く切り替えてからTraningスレッドでモデルをコピー

3. 各学習モデルは新方式にそれぞれ変更
   - 観測エンコーダ：I-JEPA
   - Forward Dynamics：SioConv
   - 強化学習方式：~~Dreamer~~　→去年と同じくPPO

2024年度版　P-AMI<Q>　大きな変更点

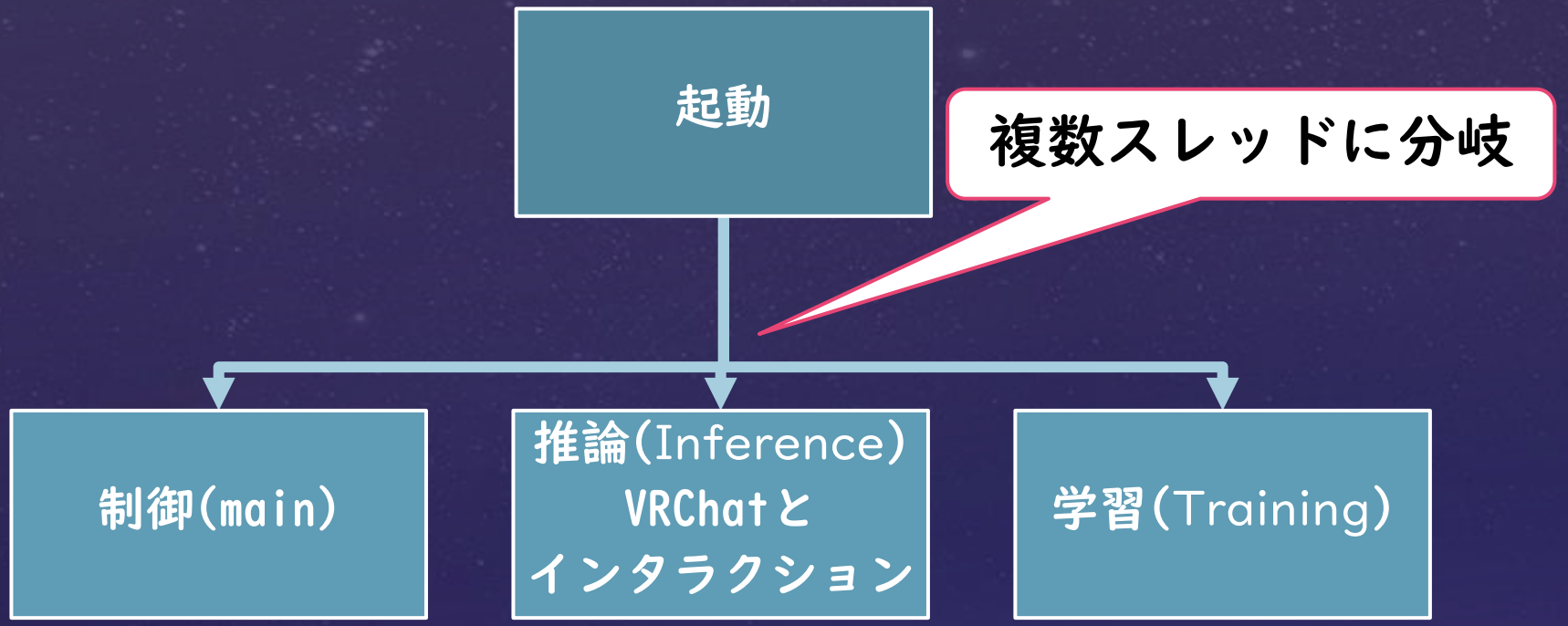
1. 処理手続きを3つのスレッドにし非同期化
   - Mainスレッド、Inferenceスレッド、Trainingスレッド

2. 推論と学習のそれぞれのデータとモデルの効
   - Inferenceスレッドで収集と推論を行いTraningスレッドに
   - 学習したモデルを素早く切り替えてからTraningスレッドでモ

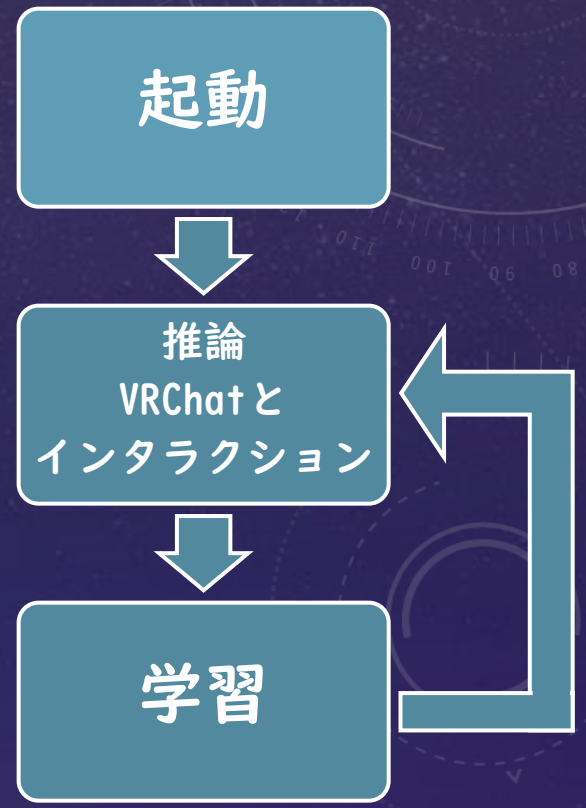
3. 各学習モデルは新方式にそれぞれ変更
   - 観測エンコーダ：I-JEPA
   - Forward Dynamics：SioConv
   - 強化学習方式：~~Dreamer~~　→去年と同じくPPO

1. 処理手続きを3つのスレッドにし非同期化

新システム



去年のシステム



2. 推論と学習のそれぞれのデータとモデルの効率的な処理

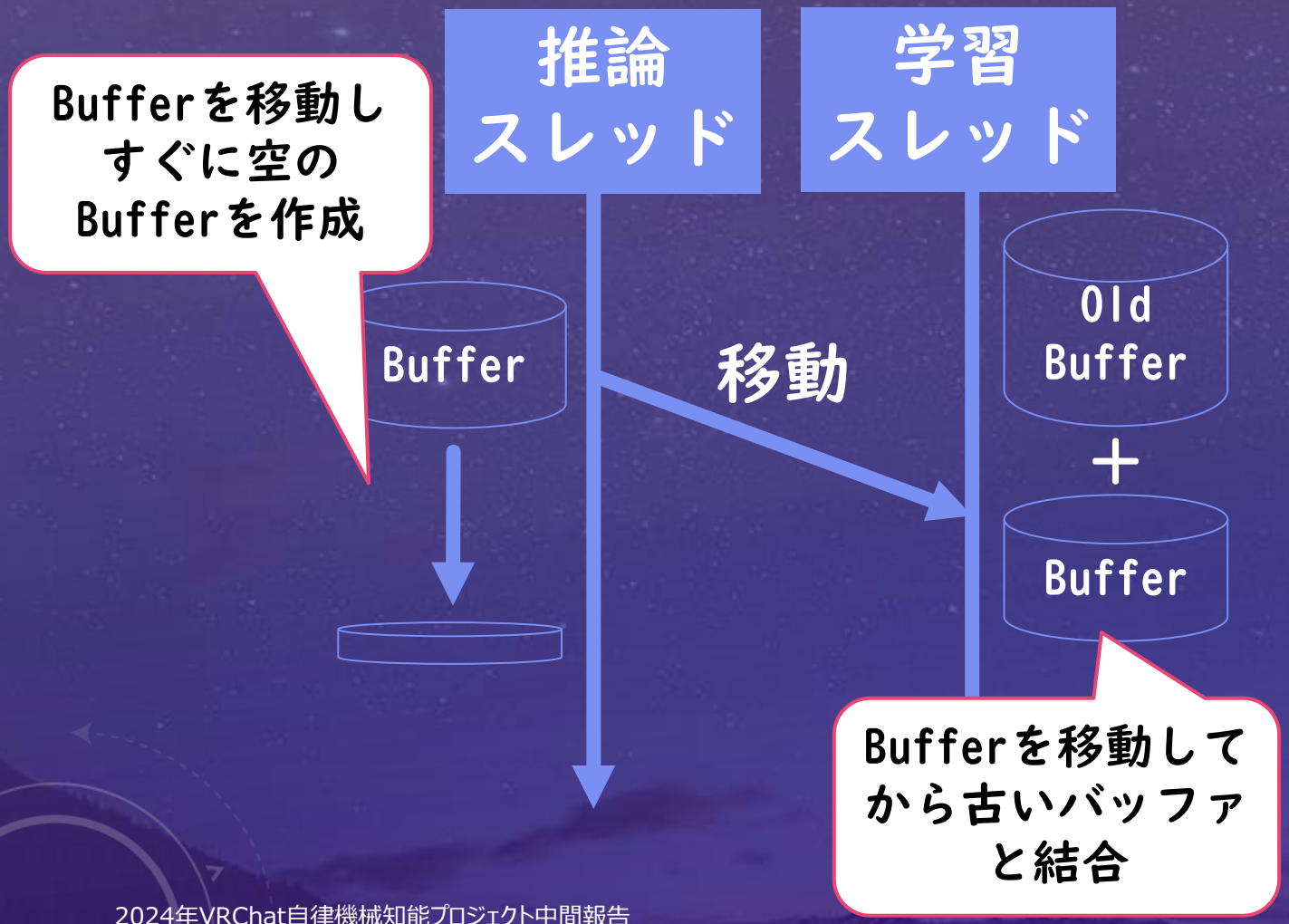
- 推論スレッドを止めてしまうとP-AMI<Q>が止まる

A) Inferenceスレッドで収集と推論を行い
Traningスレッドにデータを渡してから結合

B) **学習したモデルを素早く切り替えてから**
Traningスレッドで**古いモデル**に**新しいパラメータを同期**

といった方式で推論スレッドを止めないよう効率的に運用

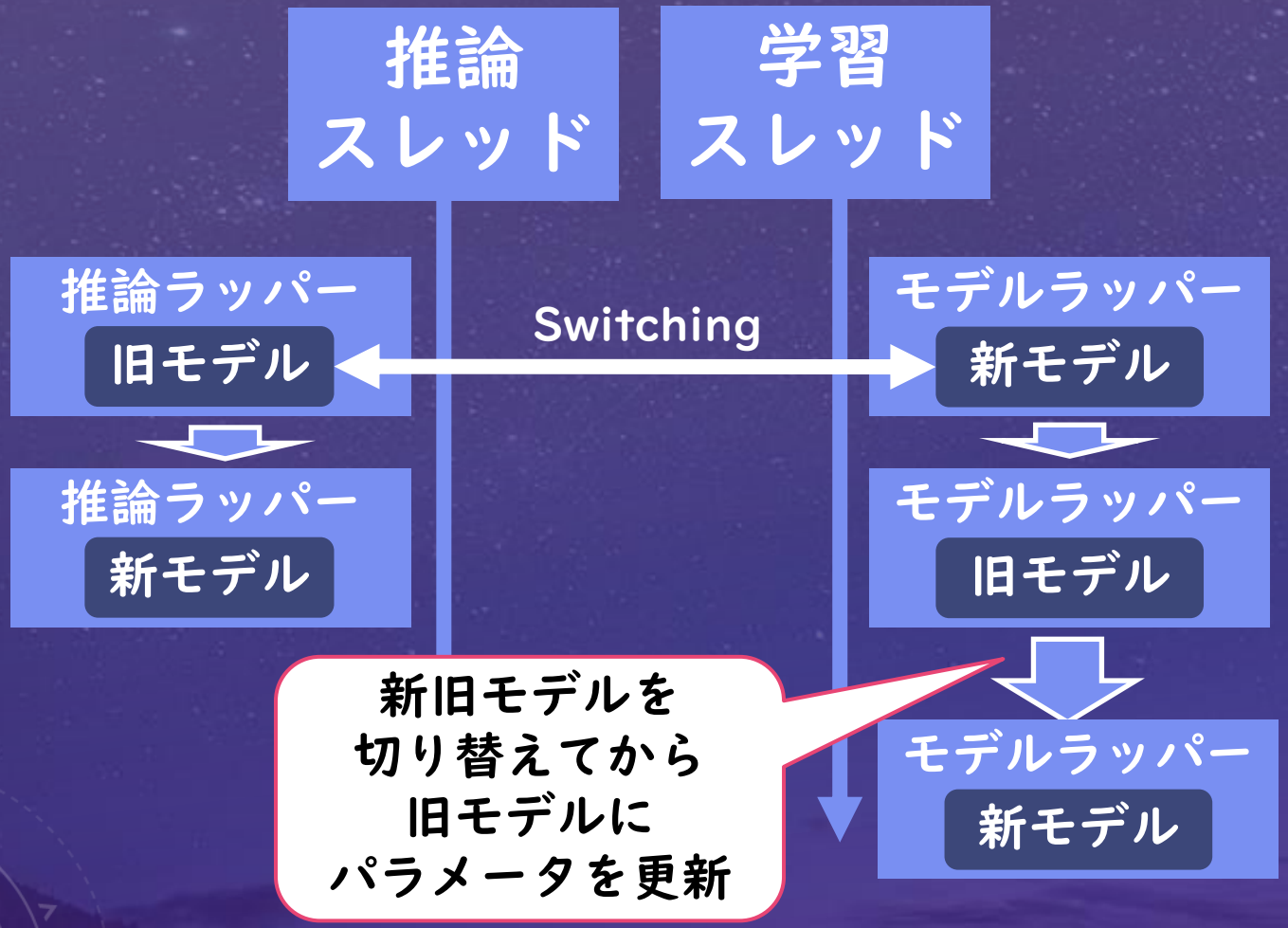
A)Inferenceスレッドで収集と推論を行い Traningスレッドにデータを渡してから結合




```python
class BaseDataBuffer(ABC):
    _init_args: tuple[Any, ...]
    _init_kwds: dict[str, Any]

    @classmethod
    def reconstructable_init(cls, *args: Any, **kwds: Any) -> Self:
        """Stores constructor arguments for renewing the data buffer, and throw
        them to :meth:`__init__`."""
        instance = cls(*copy.deepcopy(args), **copy.deepcopy(kwds))
        instance._init_args = args
        instance._init_kwds = kwds
        return instance

    @property
    def is_reconstructable(self) -> bool:
        return hasattr(self, "_init_args") and hasattr(self, "_init_kwds")

    def new(self) -> Self:
        if self.is_reconstructable:
            return self.__class__.reconstructable_init(*self._init_args, **self._init_kwds)
        else:
            raise RuntimeError(
                "Can not create new instance! Did you forget to use `reconstructable_init` "
                "instead of `__init__` when creating a instance?"
            )
```

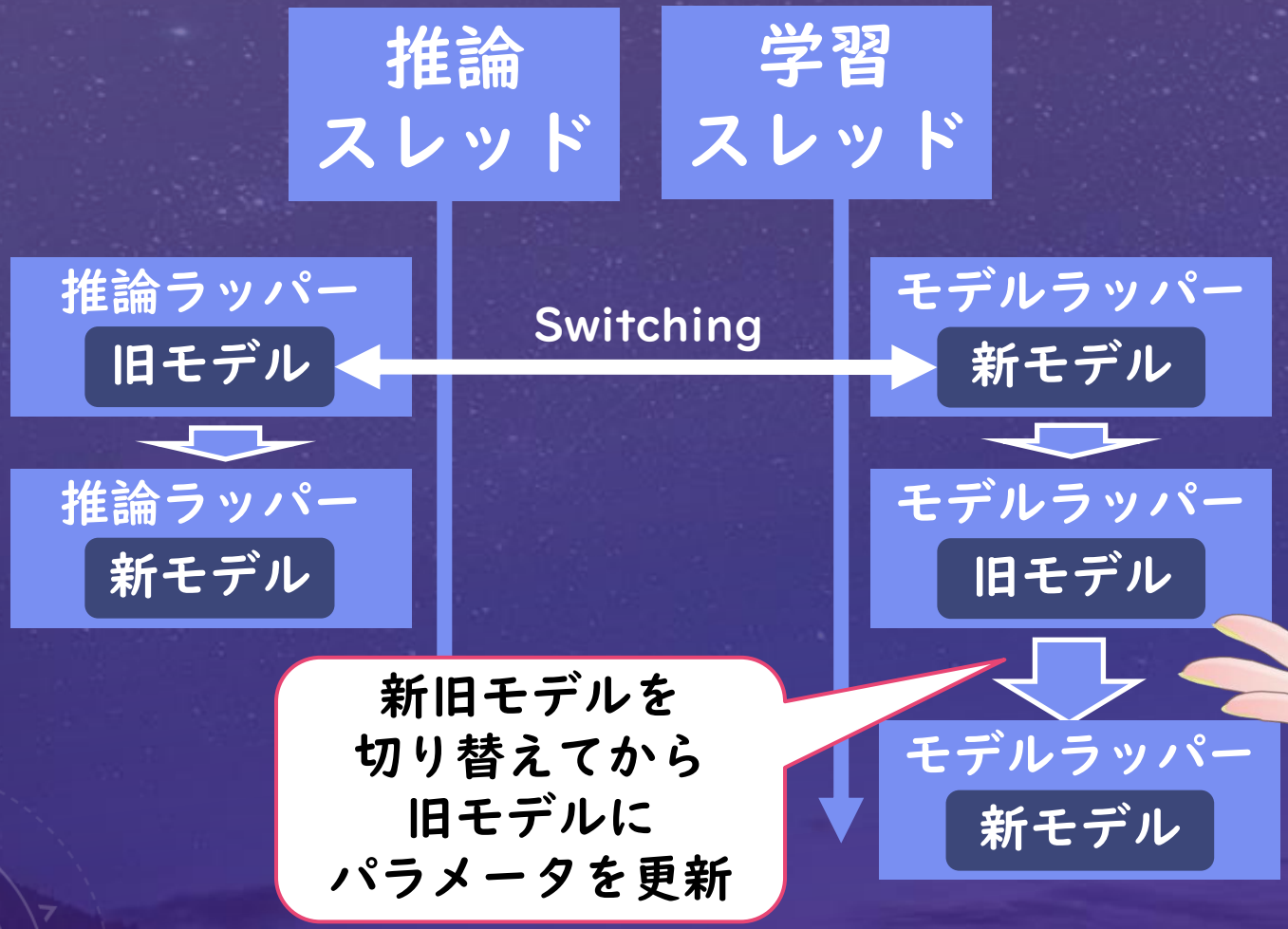
```python
class ThreadSafeDataCollector(Generic[BufferType]):
    def move_data(self) -> BufferType:
        """Move data's pointer to other object."""
        with self._lock:
            return_data = self._buffer
            self._buffer = self._buffer.new()
            return return_data
```

学習したモデルを素早く切り替えてから Traningスレッドで古いモデルに新しいパラメータを同期




```
# 学習されたモデルを推論用にセットアップ。
model_wrapper.freeze_model()

# 学習モデルと推論モデルをスイッチ。
model_wrapper.model, inference_wrapper.model = (
    inference_wrapper.model, model_wrapper.model
)       You, 1 秒前 • Uncommitted changes

# model_wrapperには推論に使われていた古いモデルが渡されるため、パラメータを同期。
model_wrapper.model.load_state_dict(inference_wrapper.model.state_dict())

# 推論に使われていたモデルを学習モードにする。
model_wrapper.unfreeze_model()
```

学習したモデルを素早く切り替えてから Traningスレッドで古いモデルに新しいパラメータを同期

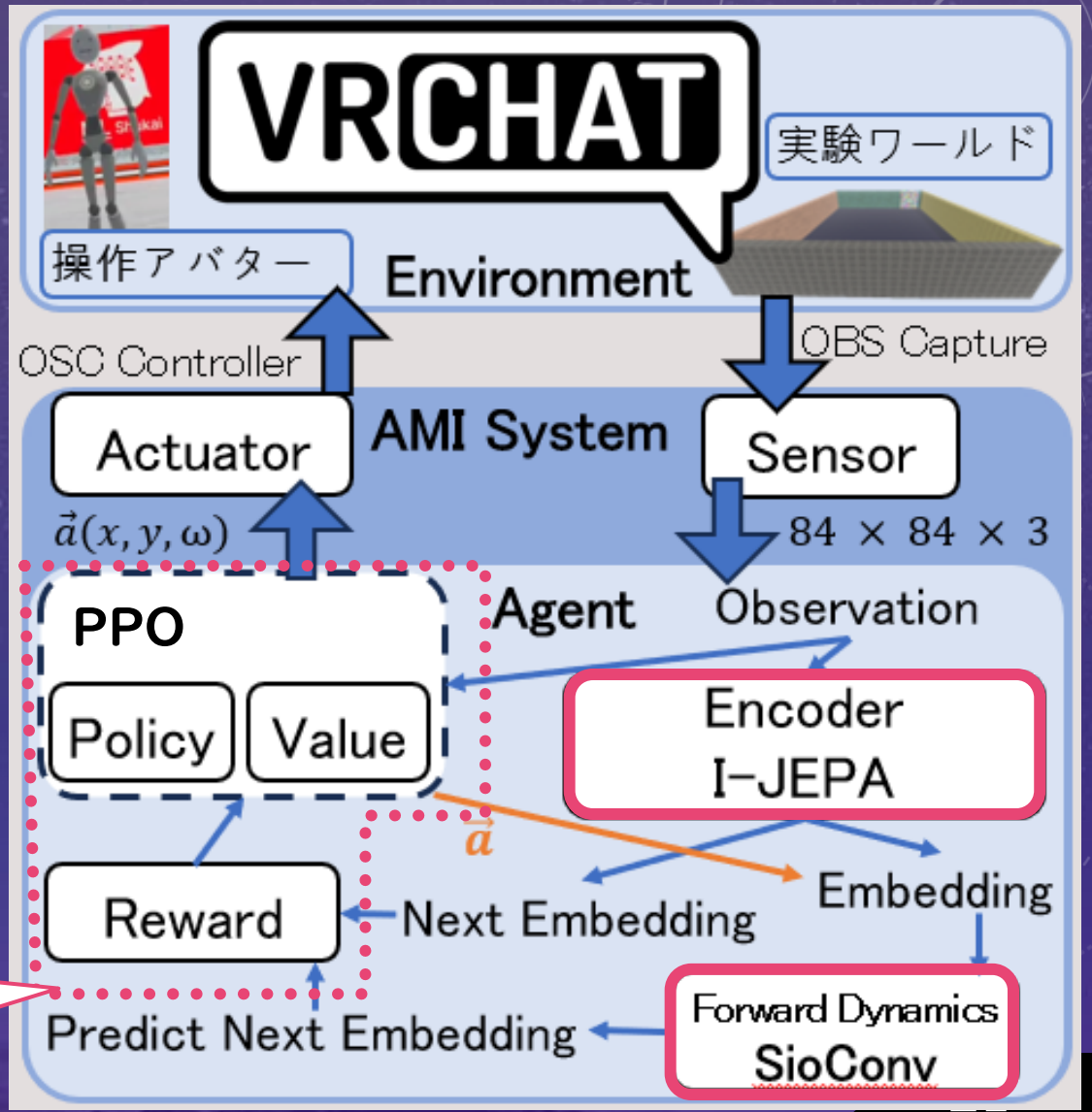



```
# 学習されたモデルを推論用にセットアップ。
model_wrapper.freeze_model()

# 学習モデルと推論モデルをスイッチ。
model_wrapper.model, inference_wrapper.model = (
    inference_wrapper.model, model_wrapper.model
)
```

どちらも推論を止めない
工夫がしてあるよ！
すごいよね！

各学習モデルは新方式にそれぞれ変更

- 観測エンコーダ：I-JEPA
- Forward Dynamics：SioConv
- 強化学習方式：~~Dreamer~~
  PPO

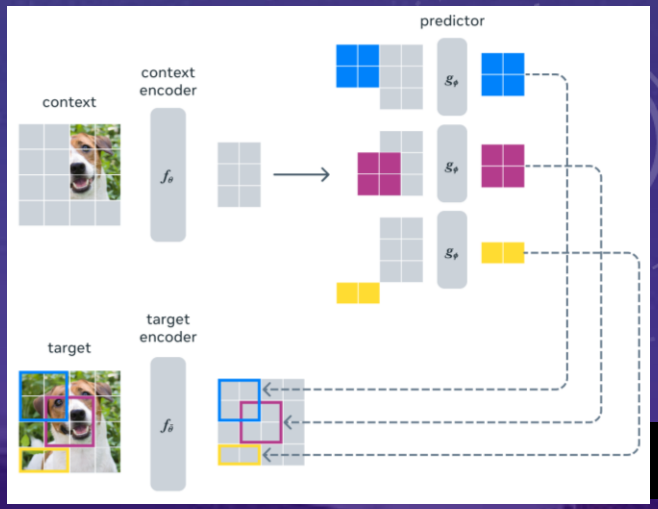


実際はマルチスレッド化されているため正確ではないが
去年のP-AMI<Q>とはそれぞれ
モデルが変わっている

観測エンコーダ：I-JEPA

- Yann LeCun氏の発表したモデル
- 画像に対する自己教師あり学習を通じて**世界の抽象的な表現を学習する**機械学習

- 詳しくはzassouさんのI-JEPAのLTを参照
  （今月末に動画化します）

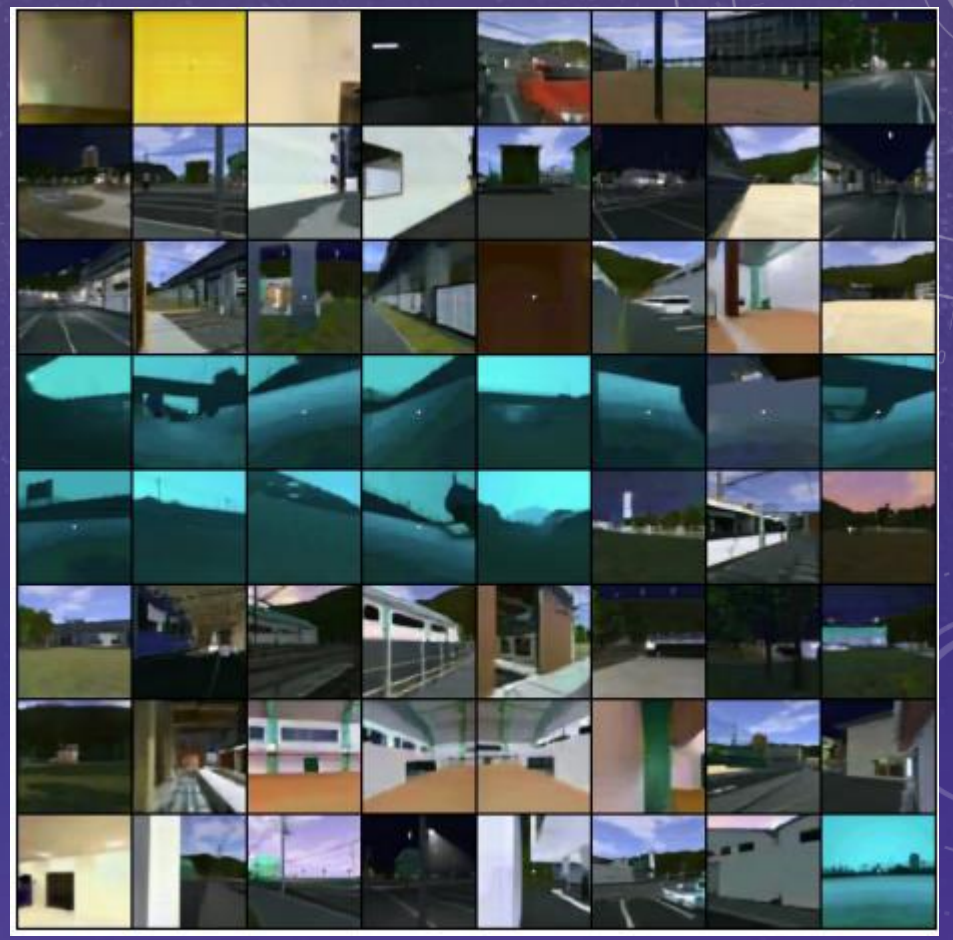


画像引用：https://ai.meta.com/blog/yann-lecun-ai-model-i-jepa/

I-JEPA修正後Decoderの再構成画像

入力画像

再構成画像

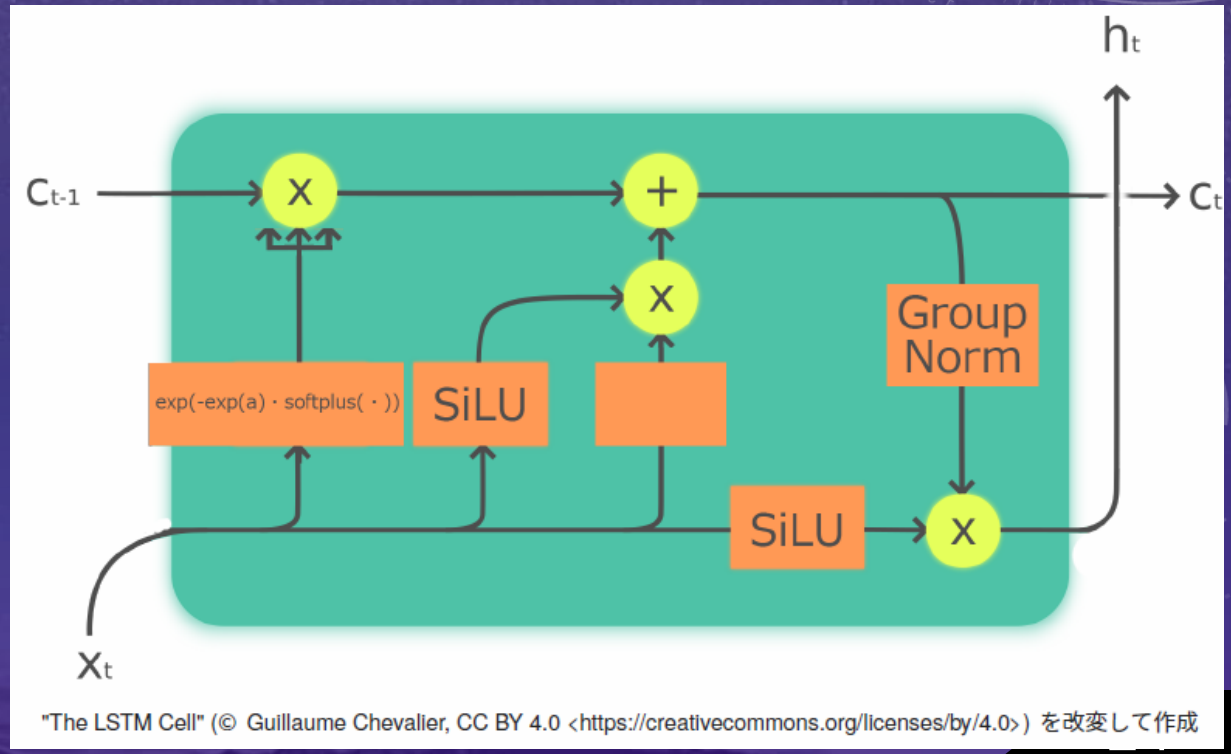


zassouさんGesonさんのI-JEPAの実験結果より

Forward Dynamics：SioConv

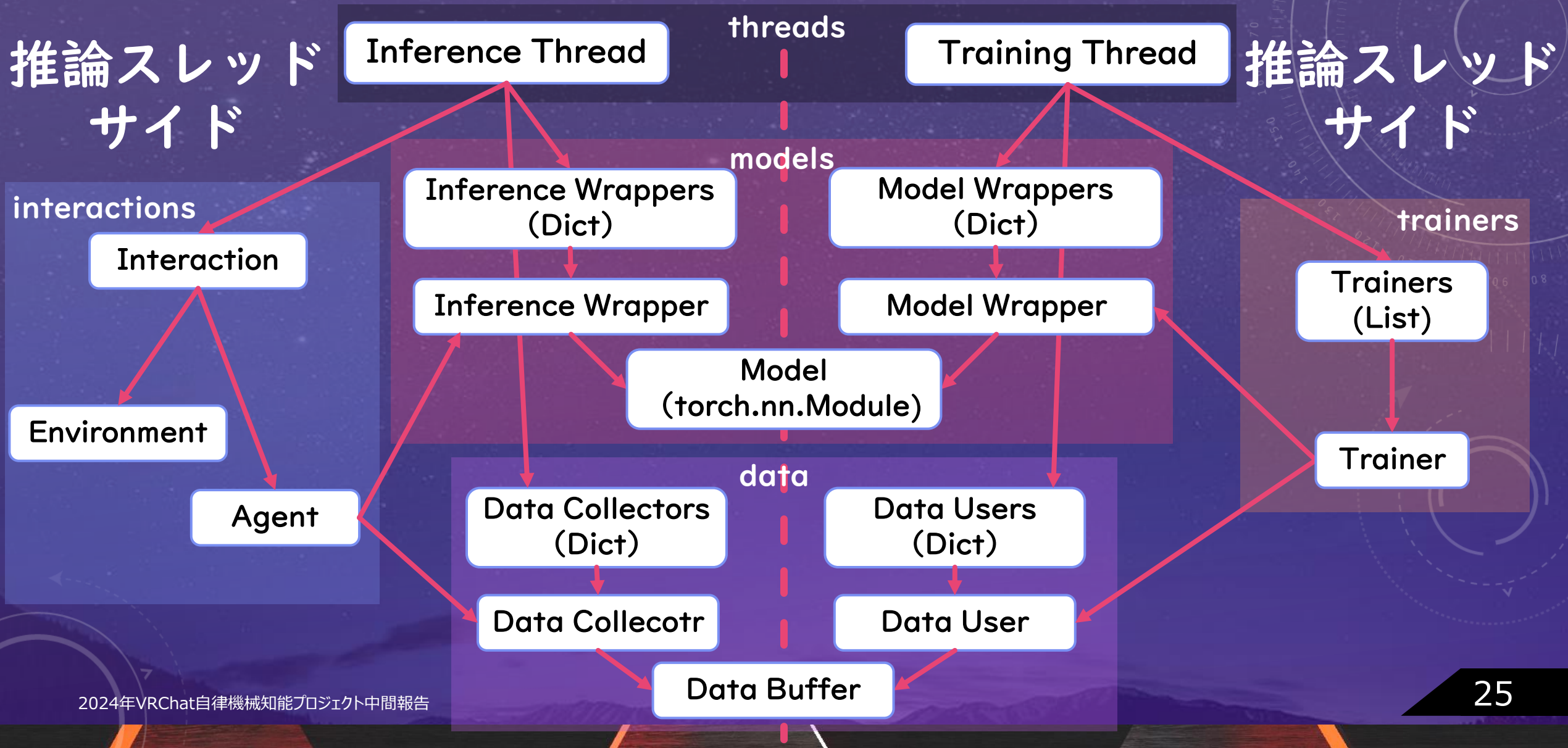
- myxyさんの開発した時系列モデル
- Transformerのように並列計算でき、RNNのように逐次的推論も出来るモデル
- **LSTM**から軽量化
- 隠れ状態を軽量化しメモリを省力化

- 詳しくはmyxyさんのLT参照
  (こちらも今月末には動画化予定)

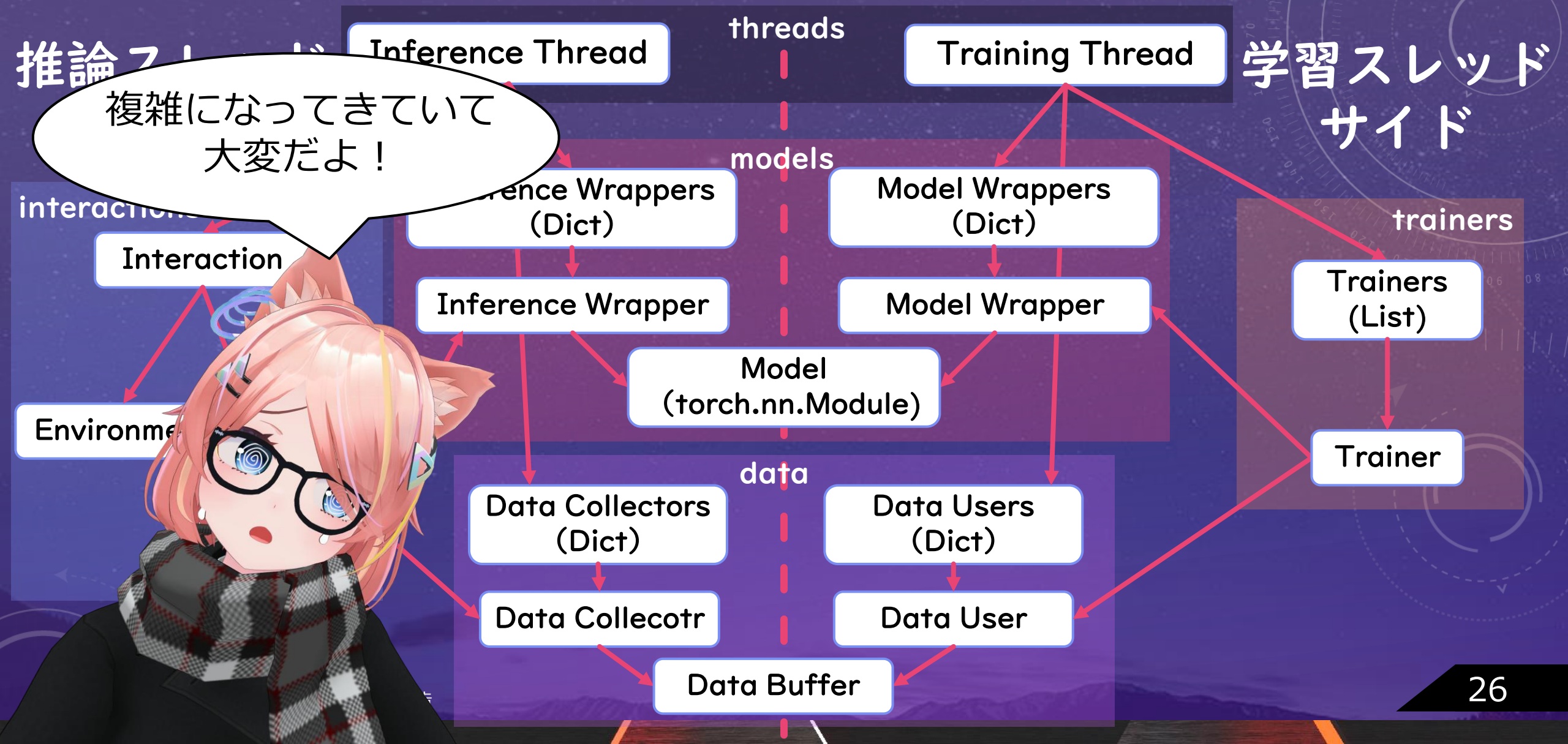


"The LSTM Cell" (© Guillaume Chevalier, CC BY 4.0 <https://creativecommons.org/licenses/by/4.0>) を改変して作成

システムの全体像（クラス関係図）



システムの全体像（クラス関係図）

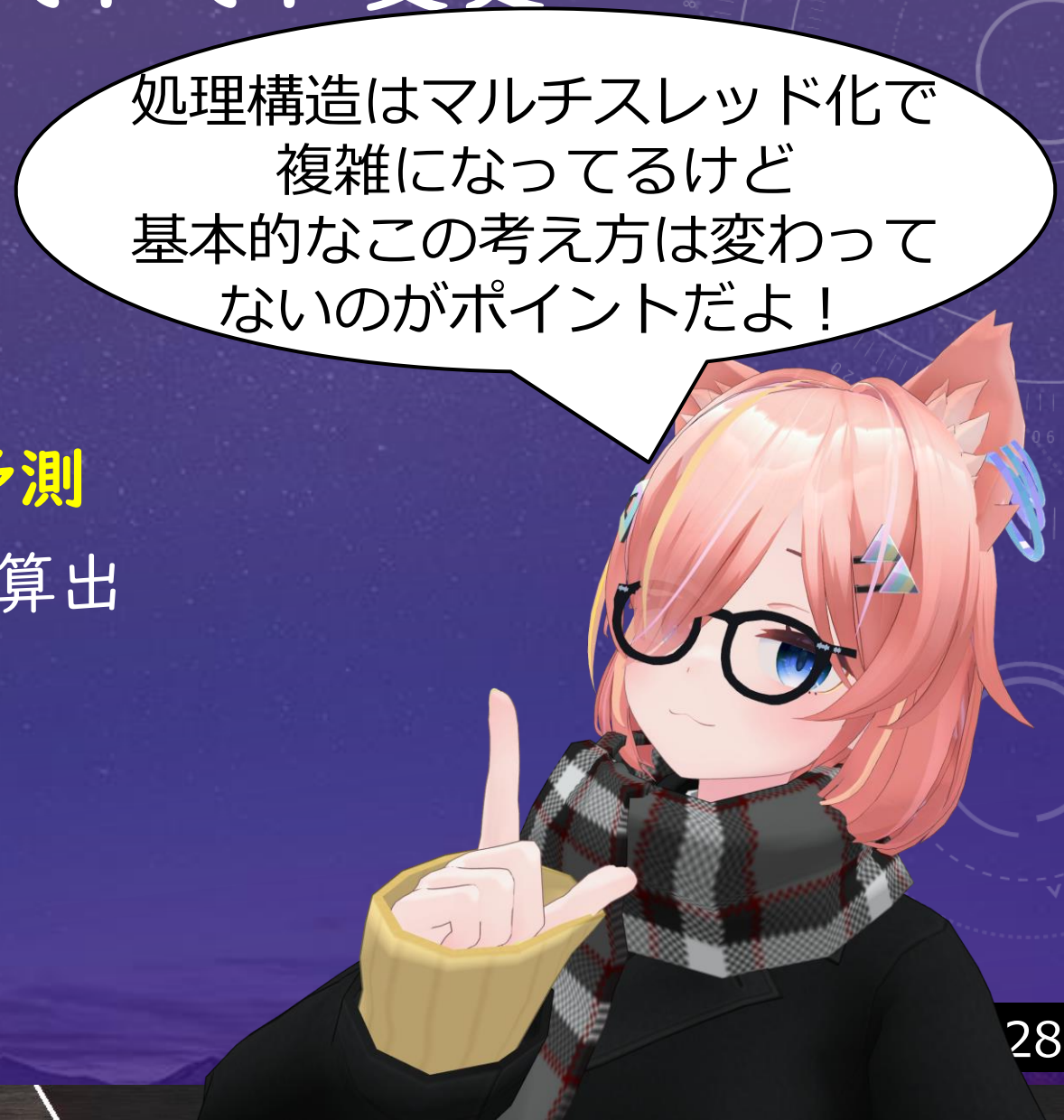


各学習モデルは新方式にそれぞれ変更

- 基本構造は内発的好奇心を報酬とした学習構造
  - 観測Encoderで**特徴量抽出**
  - Forward Dynamicsで**行動後を予測**
  - 特徴量と予測から**好奇心の報酬**を算出
  - 強化学習で次の**行動を決定**

各学習モデルは新方式にそれぞれ変更

- 基本構造は内発的好奇心を報酬とした学習構造
  - 観測Encoderで**特徴量抽出**
  - Forward Dynamicsで**行動後を予測**
  - 特徴量と予測から**好奇心の報酬**を算出
  - 強化学習で次の**行動を決定**



実際の検証機（４台）

実際の検証機（４台）

スポンサー様だよ！

- 検証機、ネットワーク環境は全てスポンサーのジー・オー・ピー株式会社様
- より提供されています

スポンサー様だよ！

- 検証機、ネットワーク環境は全てスポンサーのジー・オー・ピー株式会社様
- [illegible]れています

GOP CO.,LTD.

学習環境

- 現状は汎用的に様々なものが存在する
  Japan Streetで学習

Curiosityの課題である
本質的に予測困難な摂動
(ブロックノイズ等)
だらけのワールドで
正しく探索できるか実験予定

本質的に予測困難な摂動だらけのワールドって？

- ブロックノイズなど予測困難なもので
  埋め尽くされたワールド

- ブロックノイズだと壁と床の境界などが分かりづらい
- ランダム画像を壁と床に張り付けたワールドとして実装
- とは言え、下手な画像を使うと著作権問題が発生する

本質的に予測困難な摂動だらけのワールドって？

- ブロックノイズなど予測困難
  埋め尽くされたワールド

つまり・・・

- ブロックノイズだと壁と床の境界などが分かりづらい
- ランダム画像を壁と床に張り付けたワールドとし
- とは言え、下手な画像を使うと著作権問題が発

ランダム飯テロ画像で埋め尽くされたワールド！！

（画像の著作権保持者：Earl Klutz）

ランダム飯テロ画像で埋め尽くされたワールド!!

ランダムに秒間10枚程度
切り替わる飯テロワールドの
完成だよ！

田中スイセンさん
作ってくれて
ありがとー！

ありがとうございます

By ML集会

げそん<GesonAnko>

Myxy

Zassou

ぶんちん

田中スイセン

Earl Klutz（クルツ）