

# Guided Conditional Diffusion for Controllable Traffic Simulation

Z. Zhong et al., "Guided Conditional Diffusion for Controllable Traffic Simulation," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 3560-3566, doi: 10.1109/ICRA48891.2023.10161463.

M1 西尾舞紘

# Abstract

## 研究背景と課題

- 自動運転の開発・検証には、**現実的かつ制御可能な交通シミュレーション**が不可欠。
- 既存の手法は以下のいずれかに偏る：
  - **ヒューリスティック手法**：制御は容易だが非現実的な挙動。
  - **データ駆動手法**：挙動は人間らしいが制御が困難。

## 目的

- **制御性と現実性を両立**する新しい交通軌跡生成モデルの提案。

## 提案手法：Controllable Traffic Generation (CTG)

- **条件付き拡散モデル**を利用し、目標地点到達や速度制限などの制御を実現。
- **物理制約 + STL (Signal Temporal Logic)** で、現実性とルール制約への準拠を両立。
- **マルチエージェント対応**：相互作用を伴う複数車両間の制約（例：衝突回避）も可能。

## 検証と結果

- nuScenesで検証。多様かつ複合的な制御ルールに対して有効で**既存手法より高い制御性と現実性**を実現。

## 貢献

- 拡散モデルと論理制約の統合により、制御性・現実性・汎用性を兼ね備えた自動運転向け交通シミュレーションを実現。

# 新規性・信頼性・有用性

## 新規性

- 交通軌跡生成における「制御可能性（controllability）」を初めて定式化
- STL（Signal Temporal Logic）に基づくルール満足度を、拡散モデルのガイド関数として導入
- 車両ダイナミクス（unicycle model）を状態空間に組み込み、物理的に妥当な軌跡生成を実現
- マルチエージェントの相互作用（例：衝突回避）にも対応可能な手法を提案

## 有用性

- 実行時（inference time）に目標を指定できる柔軟な軌跡生成モデル
- 将来的な応用として、事故シナリオの生成、交通ルールのテストなどへの拡張が可能
- 衝突・逸脱などの「安全性ルール」を明示的に制御できるため、AV開発における実装可能性が高い

## 信頼性

- 大規模実データセット nuScenes を用いて評価し、既存手法と比較して高い性能を確認
- 種類の厳格なルール（速度制限、目標速度、ゴール到達、衝突回避、道路逸脱なし）に基づく定量評価
- 単一ルールおよび複数ルールの両設定で、CTGが最も低いルール違反率と安定した現実性を達成

# **1. Introduction**

# 1.1 背景

## 背景

- 交通シミュレーションは、自動運転車（AV）の開発に不可欠。現実での実験はコスト・危険性の面で困難。
- 現実性（realism）と制御可能性（controllability）の両立が鍵。

## 従来手法の限界

- 既存のAVシミュレータは多くが、記録済み走行ログの再生、ヒューリスティック制御の使用  
→人間らしさやインタラクションに欠ける（他者の動きに反応できない、意図的な操作が困難）
- 大規模データセットから挙動を生成可能（データ駆動）で、訓練分布の再現性は高いが、制御できない  
→実用場面では柔軟性が低い

## 目標

- **現実性と制御可能性のギャップを埋めること**
- **推論時に指定されたルール**（例：衝突回避・制限速度順守）を満たすような軌跡を生成
- 新しい目標・ルールにも対応可能な**柔軟な行動生成モデル**を実現

# 1.2 手法

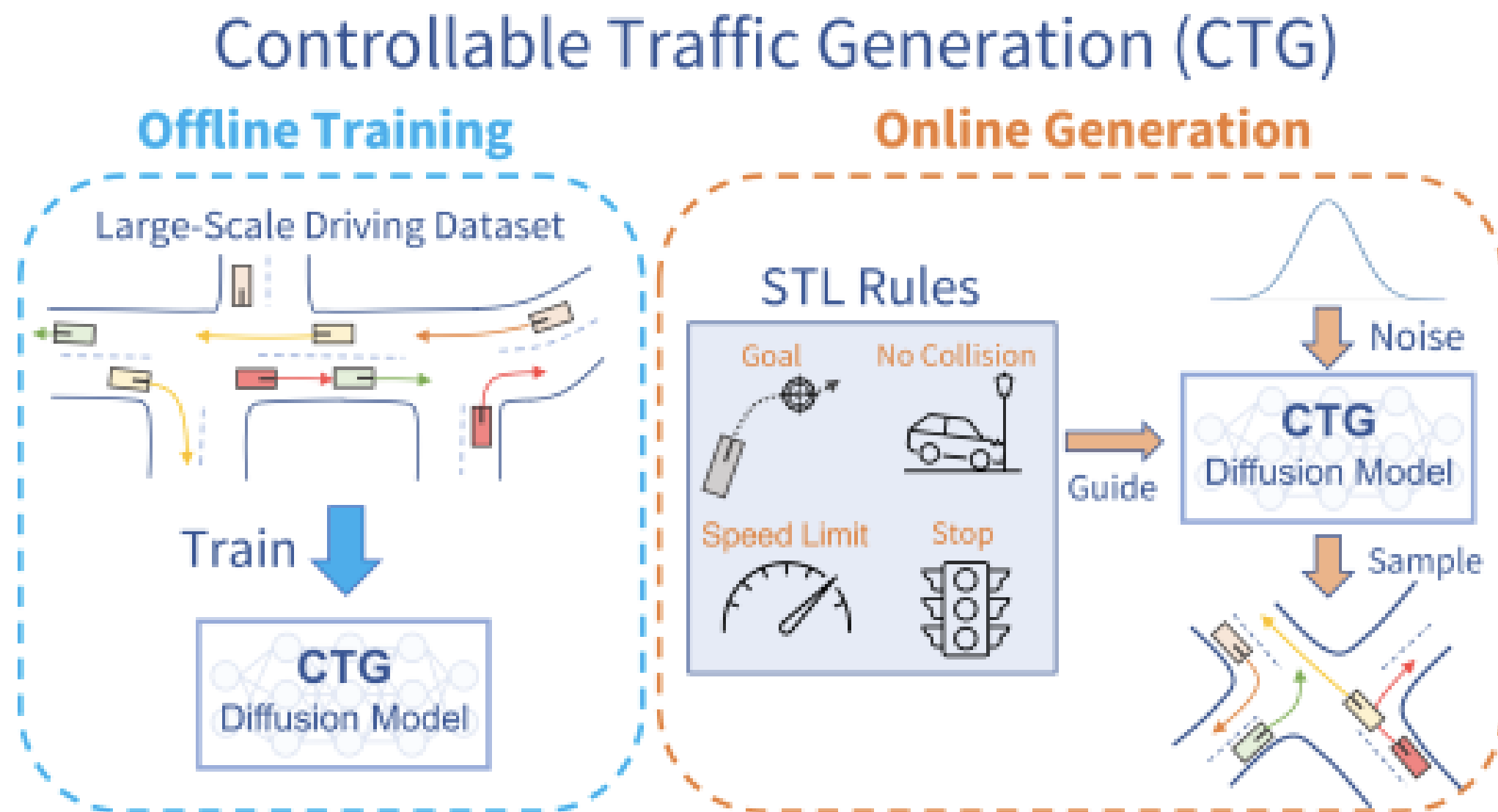
## 手法 (Controllable Traffic Generation, CTG)

### ① Offline Training (オフライントレーニング)

- 大規模運転データを用いて、**現実的な軌跡を生成する条件付き拡散モデル**を学習。
- 車両の状態・周辺状況を条件にし、現実性 (realism) を反映。

### ② Online Generation (オンライン生成)

- STLルール (例：速度制限、ゴール到達、衝突回避など) を用いて、ユーザー指定の目標に基づきサンプリング。
- STLルールの満足度 (robustness) をもとに、**拡散プロセス中の出力を誘導 (ガイド)** して制御可能性 (controllability) を実現。



# 1.2 貢献

## ① 制御可能な交通軌跡生成問題の定式化

- STLルールを用いた交通軌跡の制御可能性（Controllability）の枠組みを新たに提案。
- 複雑な交通ルール（速度制限、衝突回避、ゴール到達など）に対応可能な定式化。

## ② 拡散モデルとSTL誘導の統合

- STLルールのロバストネス関数を損失に組み込むことで、拡散モデルによる逐次生成過程において、物理的妥当性（ダイナミクス）とルール遵守の両立を実現。

## ③ 制御可能性と現実性の両立に関する実証評価

- 複数ベースラインとの比較により、CTGがルール違反率の低さと高い現実性をバランスよく実現できることを示した。
- アブレーションスタディを通じて、各要素（ガイド、フィルタリング、物理モデルなど）の効果を明確化。

## 2. Background

# 2.1 Traffic Simulation

## 1. 従来の手法

| 手法  | ルールベース                                                                   | 学習ベース                                                                      |
|-----|--------------------------------------------------------------------------|----------------------------------------------------------------------------|
| 代表例 | セルオートマトン、IDM                                                             | TrafficSim、BITS                                                            |
| 特徴  | <ul style="list-style-type: none"><li>縦横運動を個別に処理</li><li>固定ルート</li></ul> | <ul style="list-style-type: none"><li>軌跡データに基づく模倣</li><li>行動の柔軟性</li></ul> |
| 課題  | 表現力が乏しく、<br>行動の多様性に欠ける                                                   | 推論時に挙動の性質を<br>ユーザーが制御できない                                                  |

# 2.1 Traffic Simulation

## 2. 制御可能なシミュレーションの研究

**敵対的 (adversarial)** : 意図的に誤作動を起こさせる軌跡

**安全性検証 (safety-critical)** : 事故リスクのあるシナリオでの性能確認

制御可能な  
シミュレーション  
として近年注目

- Rempe et al (2022)  
訓練済みの軌跡予測モデルの潜在空間をテスト時に最適化することで、衝突寸前の危険なシナリオを生成する。
- Abeyesirigoonawardena et al. (2019)  
ベイズ最適化を用いて、交差点進入時などの高リスクな状況で敵対的軌跡を生成
- Chen et al. (2021)  
強化学習により、車線変更時に自動運転車を誤作動させるような軌跡を学習・生成。

これらは「特定の敵対的生成に特化」しており、**汎用的な目的に対応できない**点が課題

本研究の貢献：より**汎用的な制御**を実現。**複数の異なる目的に柔軟に対応可能**

# 2.2 Diffusion Modeling

拡散モデル：ノイズからデータを少しずつ「復元」していく生成手法

## 1. 拡散モデル制御法

| 手法       | 分類器ガイド                             | 分類器なしガイド                    | 再構成ガイド                    |
|----------|------------------------------------|-----------------------------|---------------------------|
| 仕組み      | 外部の分類器で生成中の出力を評価し、望ましいクラスに近づくように誘導 | 条件ありなし両方の出力を学習しその差で条件の影響を強調 | 入力（構造や目標）と一致するように損失を使って誘導 |
| 精度・誘導の強さ | ◎                                  | △                           | ◎                         |
| 実装の簡単さ   | ×<br>外部モデルが必要                      | ◎<br>モデル一つで完結               | △<br>損失設計が必要              |
| 柔軟な条件対応  | △                                  | ◎                           | ◎<br>複合条件に強い              |
| 複雑な構造・制約 | △<br>損失関数で補える場合あり                  | △<br>明示的制約には不向き             | ◎<br>複雑な制約も反映できる          |
| 手法       | 画像映像言語生成(Li et al.)                | 画像・映像生成                     | ロボット計画<br>歩行者予測、交通軌跡生成    |

# 2.2 Diffusion Modeling

## 1. 採用手法：拡散モデル再構成ガイド型

### 条件付き拡散モデルを活用

- 地図情報や周囲のエージェントの状態を入力条件に使用
- 現実的な文脈に沿った交通軌跡の生成が可能

### 車両ダイナミクスを導入

- 物理モデルに基づく運動制約を加える
- 加速度・旋回半径などを満たす物理的に妥当な軌跡を保証

### STL(Signal Temporal Logic)ルールによる明示的な制御を導入（新規性）

- 時間や論理制約を含む柔軟な条件指定が可能

| 手法       | 再構成ガイド                    |
|----------|---------------------------|
| 仕組み      | 入力（構造や目標）と一致するように損失を使って誘導 |
| 精度・誘導の強さ | ◎                         |
| 実装の簡単さ   | △<br>損失設計が必要              |
| 柔軟な条件対応  | ◎<br>複合条件に強い              |
| 複雑な構造・制約 | ◎<br>複雑な制約も反映できる          |
| 手法       | ロボット計画<br>歩行者予測、交通軌跡生成    |

# 2.2 Diffusion Modeling

## 3. 関連手法(Diffuser)との違い

### Diffuser

- ロボットやエージェントがどう動くかを決める方法
- 拡散モデルを使って行動シーケンスを生成  
ノイズから徐々に現実的な動き方を作る
- 複数のゴールや制約に対応できる柔軟な計画を目指す  
例：障害物を避けながら、目的地に向かうような動き
- 報酬関数で「良い行動」になるように誘導  
例：「できるだけ速くゴール」「ぶつからないように動く」など

|     | Diffuser           | 提案手法                 |
|-----|--------------------|----------------------|
| 仕組み | 報酬関数を設計・学習し最適化     | STLルールで直接制約を指定し軌跡を誘導 |
| 柔軟性 | ルール追加には再学習が必要な場合あり | 条件を変更・拡張しやすい         |

- 損失ではなく 軌跡生成後、明示的なルール（**STL**）に従った制御を行う

| 手法       | 再構成ガイド                    |
|----------|---------------------------|
| 仕組み      | 入力（構造や目標）と一致するように損失を使って誘導 |
| 精度・誘導の強さ | ◎                         |
| 実装の簡単さ   | △<br>損失設計が必要              |
| 柔軟な条件対応  | ◎<br>複合条件に強い              |
| 複雑な構造・制約 | ◎<br>複雑な制約も反映できる          |
| 手法       | ロボット計画<br>歩行者予測、交通軌跡生成    |

# 2.3 Signal Temporal Logic (STL)

## 1. STL

### Signal Temporal Logic (STL)

時系列データに対して論理的・時間的制約を与えるための形式論理

$\tau = \{s_t, s_{t+1}, \dots, s_{t+T}\}$   $s_i \in \mathbb{R}^n$  : n次元の状態ベクトル (位置、速度、角度など)

### 基本式 (STL式)

$$\phi ::= \top \mid \mu_c \mid \neg \phi \mid \phi \wedge \psi \mid \phi \mathcal{U}_{[a,b]} \psi$$

例) 常に速度が5m/s以下で、10秒以内にゴールに到達する  
 $\phi = (\text{speed}(z) < 5) \mathcal{U}_{[0,10]} \text{goal}(z)$

STL式  $\phi$  は、以下の5つの方法のいずれかで構成される (他の演算子も使用可)

- |                                                                                               |                                                                                                     |                                                                                           |                                                                                                            |                                                                                                                                                |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| ① $\phi ::= \top$                                                                             | ② $\phi ::= \mu_c$<br><small>微分可能な関数</small>                                                        | ③ $\phi ::= \neg \phi$                                                                    | ④ $\phi ::= \phi \wedge \psi$                                                                              | ⑤ $\phi ::= \phi \mathcal{U}_{[a,b]} \psi$                                                                                                     |
| <ul style="list-style-type: none"><li>「常に真である (True)」という意味</li><li>条件なしで常に成立している論理式</li></ul> | <ul style="list-style-type: none"><li>述語</li><li>状態 <math>z</math> がある条件 (距離や速度など) を満たすか?</li></ul> | <ul style="list-style-type: none"><li>否定</li><li><math>\phi</math> が成り立たないことを意味</li></ul> | <ul style="list-style-type: none"><li>論理積</li><li><math>\phi</math> と <math>\psi</math> の両方が成り立つ</li></ul> | <ul style="list-style-type: none"><li>Until (～まで)</li><li><math>\phi</math> が成立し続け、<math>a \sim b</math>秒以内に <math>\psi</math> が成り立つ</li></ul> |

# 2.3 Signal Temporal Logic (STL)

## 1. STL

### 基本式 (STL式)

$$\phi ::= \top \mid \mu_c \mid \neg\phi \mid \phi \wedge \psi \mid \phi \mathcal{U}_{[a,b]} \psi$$

例) 常に速度が5m/s以下で、10秒以内にゴールに到達する  
 $\phi = (\text{speed}(z) < 5) \mathcal{U}_{[0,10]} \text{goal}(z)$

STL式  $\phi$  は、以下の5つの方法のいずれかで構成される (他の演算子も使用可)

- |                                                                                               |                                                                                                     |                                                                                           |                                                                                                            |                                                                                                                            |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| ① $\phi ::= \top$                                                                             | ② $\phi ::= \mu_c$                                                                                  | ③ $\phi ::= \neg\phi$                                                                     | ④ $\phi ::= \phi \wedge \psi$                                                                              | ⑤ $\phi ::= \phi \mathcal{U}_{[a,b]} \psi$                                                                                 |
| <ul style="list-style-type: none"><li>「常に真である (True)」という意味</li><li>条件なしで常に成立している論理式</li></ul> | <ul style="list-style-type: none"><li>述語</li><li>状態 <math>z</math> がある条件 (距離や速度など) を満たすか?</li></ul> | <ul style="list-style-type: none"><li>否定</li><li><math>\phi</math> が成り立たないことを意味</li></ul> | <ul style="list-style-type: none"><li>論理積</li><li><math>\phi</math> と <math>\psi</math> の両方が成り立つ</li></ul> | <ul style="list-style-type: none"><li><math>\phi</math> が成立し続け、<math>a \sim b</math>秒以内に <math>\psi</math> が成り立つ</li></ul> |

論理式の「満たす／満たさない」に加え、どの程度満たしているかを数値で表す **ロバスト性** が定義されている。

論理式の満足度を連続値のスコアとして評価可能。

ロバスト性を **損失関数として活用** し、拡散モデルの生成軌跡を論理条件に沿うよう最適化

# **3. CONTROLLABLE TRAFFIC GENERATION**

# 3.1 Problem Formulation

## ① 交通エージェントの状態を定義

$s_t^{tgt} = (x_t, y_t, v_{x,t}, v_{y,t}, \theta_t)$

位置                  速度                  ヨー角（車両の向いている角度）

## ② 対応する制御（行動）を定義

$a_t^{tgt} = (a_t, \delta_t)$

加速度          ヨーレート（車の曲がる速さ）

## ③ コンテキスト（意思決定のための情報）を定義

$c = (I, S)$

意味地図（semantic map）  
例：車線、障害物、信号など

周囲の状態履歴。対象車両と周囲の M 台の車両の、H ステップ分の過去の状態：  
 $S_{t-H:t} = \{s_{t-H:t}^{tgt}, s_{t-H:t}^1, \dots, s_{t-H:t}^M\}$

## ④ 次の状態の推定方法：力学モデルを使う

車両  $m$

$s_{t+1}^m = f(s_t^m, a_t^m)$

unicycle model（一輪車モデル）：  
方向を持った車両の動きを表す基本的な  
動力学モデル

## ⑥ モデルの出力

対象車両の未来の状態列

$s_{t+1:t+T}^{tgt}$  ← 軌跡の質を測る

ルール満足度スコア

$r(s_{t+1:t+T}^{tgt}, a_{t+1:t+T}^{tgt})$

目標

現実的かつルールを守った交通挙動を、  
あるコンテキスト下で生成すること

不自然な動きになる可能性

制約の満足と挙動の現実性のバランスを  
取るように設計

# 3.2 Conditional Diffusion for Traffic Modeling

## 1. Trajectory representation

将来の軌跡（未来の状態と行動のシーケンス）をどのように表現・生成するかを定式化

### 軌跡の定義

拡散モデルは 行動のみ生成

$\tau = \begin{matrix} \boxed{\tau_a} \\ \boxed{\tau_s} \end{matrix}$  行動列  $\tau_a = [a_0, a_1, \dots, a_{T-1}]$   
状態列  $\tau_s = [s_1, s_2, \dots, s_T]$

### 正方向拡散（ノイズを加えるプロセス）

$$q(\tau_a^{1:K} | \tau_a^0) = \prod_{k=1}^K q(\tau_a^k | \tau_a^{k-1})$$
  
きれいなデータ  $\tau_a^0$  から最後のノイズ付きデータ  $\tau_a^K$  までの確率的な変化を掛け合わせた式

各ステップの式

$$q(\tau_a^k | \tau_a^{k-1}) = \mathcal{N}(\tau_a^k; \sqrt{1 - \beta_k} \tau_a^{k-1}, \boxed{\beta_k I})$$
  
注入されるノイズ量

### 復元した行動列から状態を再構成

状態は物理的な遷移モデル  $f$  によって求められる

履歴から与えられる  
初期状態

逆拡散で生成  
された元の行動列

$\tau_s = f(\boxed{s_0}, \tau_a)$  再構成  $\tau_s = f(\boxed{s_0}, \boxed{\tau_a^0})$

### 逆拡散（ノイズから軌跡を再構成）

$$p_\theta(\tau_a^{0:K} | c) = p(\tau_a^K) \prod_{k=1}^K p_\theta(\tau_a^{k-1} | \tau_a^k, \boxed{c})$$
  
条件

各ステップの式

$$p_\theta(\tau_a^{k-1} | \tau_a^k, c) = \mathcal{N}(\boxed{\mu_\theta}(\cdot), \Sigma_\theta(\cdot))$$
  
学習済みモデルで近似された条件付き分布

# 3.2 Conditional Diffusion for Traffic Modeling

## 2. Training (オフライン)

### 教師データの準備

毎回の学習イテレーションで、データセットからランダムにサンプリング

- コンテキスト  $c$
- ノイズのない軌跡  $\tau^0$

### ノイズステップ $k$ のサンプリング

正方向拡散のステップ数  $k$  を、 $1 \sim K$  の中からランダムに選ぶ  
(学習時に毎回異なるレベルのノイズを使うため)

### 正方向拡散

ノイズがない行動列  $\tau_a^0$  にノイズを加えてノイズ付き  $\tau_a^k$  を生成

$$\tau_a^k = \sqrt{\bar{\alpha}_k} \tau_a^0 + \sqrt{1 - \bar{\alpha}_k} \epsilon \quad \epsilon \sim \mathcal{N}(0, I)$$

ノイズスケジュールの累積項  $\prod_{i=1}^k (1 - \beta_i)$

標準正規分布からのサンプル。  
ランダムノイズ

### 状態列の再計算

ノイズ付きの行動列  $\tau_a^k$  から、状態列  $\tau_s^k$  を物理モデル  $f$  によって再計算。状態は決定論的に決まる

$$\tau_s^k = f(s_0, \tau_a^k)$$

### モデルの出力 (逆拡散)

ノイズ付きの  $\tau_a^k$ 、ステップ  $k$ 、コンテキスト  $c$  を入力として、元の行動列  $\hat{\tau}_a^0$  を直接予測

$$\hat{\tau}_a^0 = \mu_\theta(\tau_a^k, k, c)$$

元の状態列  $\hat{\tau}_s^{(k)}$  も再構成、モデル出力は以下

$$\hat{\tau}^0 = [\hat{\tau}_a^0, f(s_0, \hat{\tau}_a^0)]$$

### 損失関数

モデル出力  $\hat{\tau}^0$  と正解データ  $\tau^0$  の差を評価

$$L(\theta) = \mathbb{E}_{k, \tau, c} [\|\tau^0 - \hat{\tau}^0\|^2]$$

平均二乗誤差 (MSE) に基づく損失

# 3.2 Conditional Diffusion for Traffic Modeling

## 3. Implementation details

### コンテキスト $c$

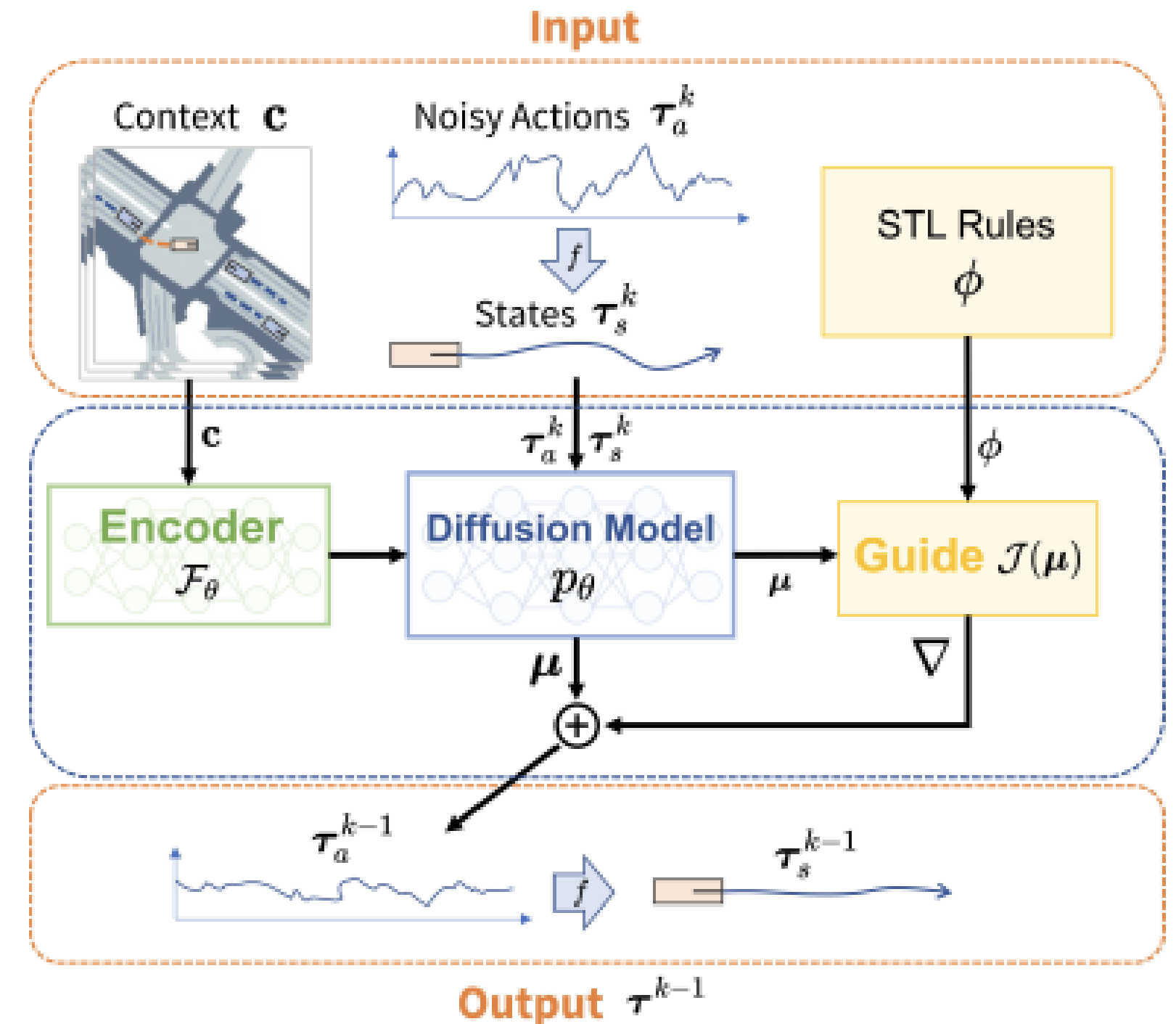
- エージェント中心のマップ情報 + 過去の軌跡データ
- ラスタ形式に変換

### ノイズ付き行動 $\tau_a^k$

- Noisy Actions
- 正方向拡散でノイズが加えられた**行動軌跡**
- 逆拡散の出発点となる入力データ

### 状態列 $\tau_s^k$

- 関数  $f$  を通じて、行動から得られる「状態軌跡」
- 動的モデル  $f(s_0, \tau_a^k)$  によって決定論的に再計算



# 3.2 Conditional Diffusion for Traffic Modeling

## 3. Implementation details

### エンコーダ $\mathcal{F}_\theta$

- ResNetベースのニューラルネットワーク
- 地図や軌跡といった画像的データをベクトル形式に変換し、拡散モデルに渡す

### 拡散モデル $p_\theta$

- 中心モデル。U-Netのような構造で、時系列軌跡に対し
- 1次元畳み込みを繰り返す。
- 入力にはコンテキスト・ノイズ付き行動・状態・ステップ  $k$

### ガイド $\mathcal{J}(\mu)$

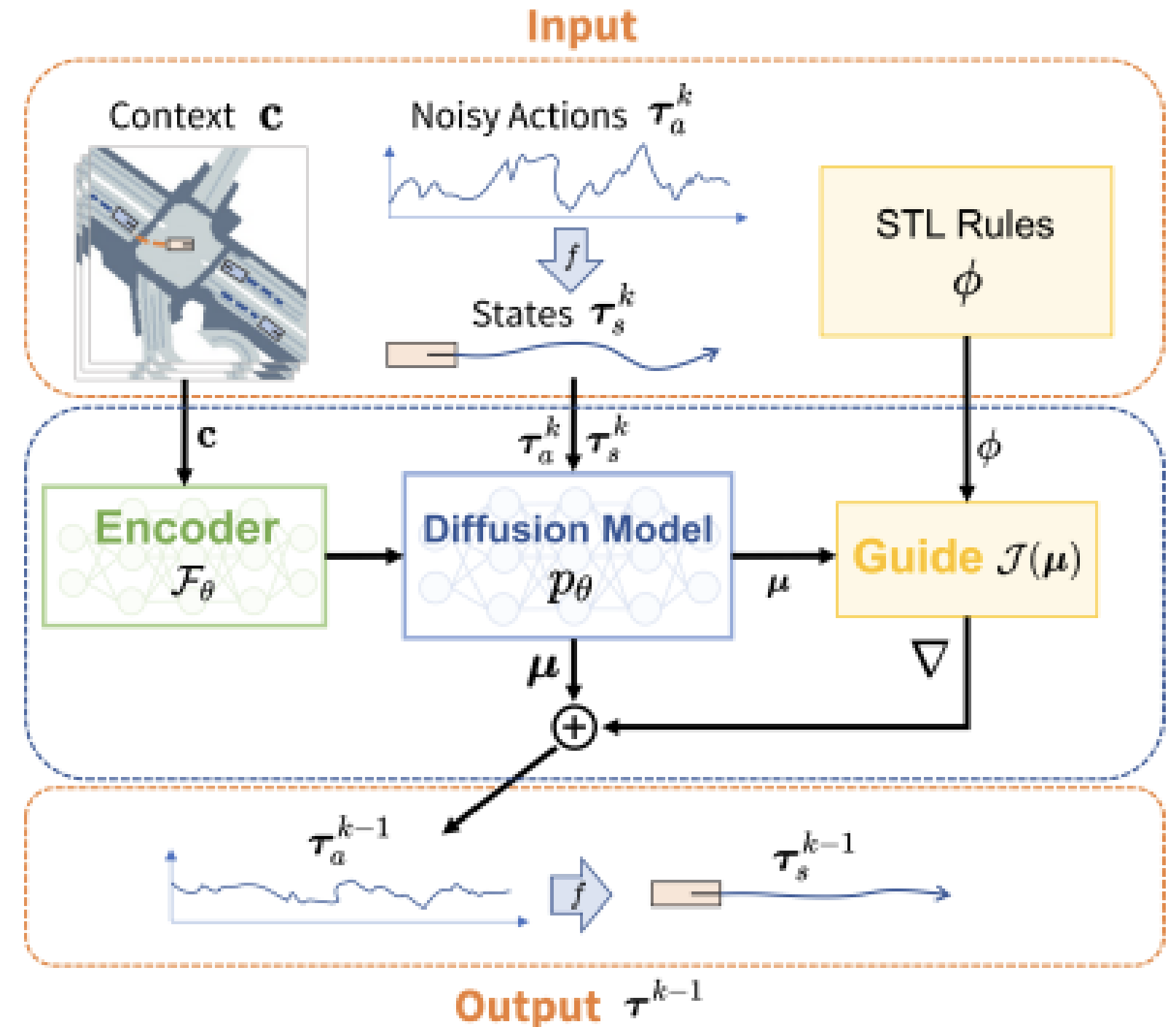
- STLルール  $\phi$  に従って、生成された出力がルールを守るように勾配を与える機構

### ノイズスケジュールとステップ数 $K$

- コサインスケジュールを使用し、全実験で  $K = 100$  ステップ

出力  $\tau_a^{k-1}, \tau_s^{k-1}$

- 再構成された行動と状態の波形
- 拡散ステップを1つ戻した結果



# 3.3 Guided Generation with Signal Temporal Logic (STL)

## 1. Conditional guidance formulation. (オンライン)

無条件拡散モデルから、ある目的を満たす軌跡を得るために、生成の途中で出力を誘導する = **ガイダンス**

ルールを満たしている軌跡の確率

ルールに準拠していること  
 $p(\mathcal{O} = 1) = \exp(r(\tau_s, \tau_a))$   
ルール満足度を評価する関数

拡散モデルの補正方法 (勾配による誘導)

条件付きガイドなしの通常出力分布

$$p_{\theta}(\tau_a^{k-1} \mid \tau^k, c) \approx \mathcal{N}(\mu_{\theta}, \Sigma_{\theta})$$

- 平均  $\mu_{\theta}$ 、共分散  $\Sigma_{\theta}$  のガウス分布からサンプリング

ルール満足度に基づく勾配補正：ルールを満たすように行動を誘導

$$p_{\theta}(\tau_a^{k-1} \mid \tau^k, c, \mathcal{O}) \approx \mathcal{N}(\tau_a^{k-1}; \mu + \Sigma g, \Sigma)$$

- ルールを満たす方向へ平均値をずらす

ルール満足度の勾配

$$g = \nabla_{\mu} \mathcal{J}(\mu) = \nabla_{\mu} r(f(s_0, \mu), \mu)$$

現在の状態列      現在の行動予測

- 行動をどう変えれば満足度が上がるかの勾配を計算

- 状態列  $\tau_s$  は、初期状態  $s_0$  と補正後の行動列から再計算：  
 $\tau_s = f(s_0, \tau_a)$

# 3.3 Guided Generation with Signal Temporal Logic (STL)

## 1. Conditional guidance formulation.

### Algorithm 1 Guided Sampling

```
1: Require encoder  $\mathcal{F}_\theta$ , conditional diffusion model  $\mu_\theta$ , transition function  $f$ , guide  $\mathcal{J}$ , scale  $\alpha$ , covariances  $\Sigma^k$ , diffusion steps  $K$ , inner gradient descent steps  $M$ , number of actions to take before re-planning  $l$ .
2: while not done do
3:   Observe state  $s_0$  and context  $\mathbf{c}$ 
4:   Initialize trajectory  $\tau_a^K \sim \mathcal{N}(\mathbf{0}, I)$ ;  $\tau_s^K = f(s_0, \tau_a^K)$ ;  $\tau^K = [\tau_a^K; \tau_s^K]$ 
5:   for  $k = K, \dots, 1$  do
6:      $\mu = \mu_\theta(\tau^k, k, \mathcal{F}_\theta(\mathbf{c}))$ 
7:      $\mu^{(0)} = \mu$ 
8:     for  $j = 1, \dots, M$  do
9:        $\mu^{(j)} = \mu^{(j-1)} + \alpha \nabla \mathcal{J}(\mu^{(j-1)})$ 
10:       $\Delta\mu = |\mu^{(j)} - \mu^{(0)}|$ ;  $\Delta\mu \leftarrow \text{clip}(\Delta\mu, -\beta_k, \beta_k)$ 
11:       $\mu^{(j)} \leftarrow \mu^{(0)} + \Delta\mu$ 
12:       $\tau_a^{k-1} \sim \mathcal{N}(\mu^{(M)}, \Sigma^k)$ ;  $\tau_s^{k-1} = f(s_0, \tau_a^{k-1})$ ;
         $\tau^{k-1} = [\tau_a^{k-1}; \tau_s^{k-1}]$ 
13:   Execute first  $l$  actions of trajectory  $\tau_a^0$ 
```

8～11行目

- 勾配更新を  $M$  回繰り返す

Diffuser との大きな違い( $j=1$ )

- Adam最適化器を使って学習率や安定性を向上。
- clip処理で補正の暴走を防ぐ。
  - 勾配更新の量が大きくなりすぎないように

ルール順守と軌道の自然さのバランスが取りやすくなる。

### フィルタリング

拡散モデルは確率的に複数の軌跡を生成する。  
各軌跡に対して、ルール満足度スコア  $\mathcal{J}(\mu)$  を計算。

最もスコアが高い軌跡だけを選び、  
最終出力の質を高める。

# 3.3 Guided Generation with Signal Temporal Logic (STL)

## 2. STL as guidance

**STLを使ってルールを明示的に記述**し、そのルールの満足度を評価する数式（ロバストネス関数）をガイド関数 $J$ として利用

STLを使う理由

- 分類器や報酬関数を学習せずに済む。
- STL式は設計が簡単で人間が意味を持たせやすい。
- 微分可能な実装ができれば、勾配を通じてモデルを誘導できる。

## 3. Multi-agent guidance

- 衝突回避など複数エージェントをまたぐルールに対応
- 全エージェントを一括で処理し、全体満足度を計算
- 勾配を各エージェントに分配し、協調的な行動を誘導

## 4. Simulating traffic

- 同じモデルを各エージェントに共通適用（エージェントごとに個別学習しない）
- 拡散モデルで各エージェントの将来行動を逐次生成

# 4. Experiments

# 4.1 Experimental Setup

## 1. 使用データセット

- **nuScenesデータセット**を使用
  - 実世界の都市走行ログを元に構成された大規模なデータセット
  - 合計5.5時間分の走行軌跡を収録
  - 多様なシナリオ（例：交差点、混雑、直進など）と車両密度を含む2つの都市から収集
  - 訓練セットと検証セットに分割し、評価には検証セットから100シーンをランダム抽出
  - **車両のシミュレーションのみ**に注力しており、歩行者・自転車などの対象は今後の課題

## 2. 評価指標

- **制御性**
  - STLルールをどの程度満たしているかを評価
  - 各車両に対して個別に計算し、各シーンの平均をとる
- **現実性**
  - 実データと生成データの分布をWasserstein距離で比較
  - 平均絶対誤差でどれくらい現実の運転らしいかを数値で評価
    - 対象：縦加速度、横加速度、ジャーク（加加速度）
- **安定性**
  - 衝突や道路逸脱などの重大失敗の平均発生率を計測

# 4.1 Experimental Setup

## 3. Baselines

「ルール準拠型の交通軌跡生成」というタスクは新しいため、既存の手法とそのまま直接比較できるベースラインが存在しない

既存の代表的な交通モデルに対して生成後にルールを満たすよう補正を追加することで、ルール誘導を行えるようにする

### SimNet+opt

- 訓練データの行動をそのまま模倣する決定論的模倣学習モデル
- 生成された行動に TTO (test-time optimization) を適用してルールに沿うよう補正

### TrafficSim+opt

- CVAEに基づく軌跡生成モデル
- ルールに沿った行動を得るために：
  - フィルタリング：複数サンプルから良いものを選ぶ
  - 潜在空間の最適化：内部表現をルール適合に合わせて調整

### BITS+opt

- 候補生成 + ランキングの2段階からなる bi-level模倣学習モデル
- ランキング関数に ルール満足度 (STLガイド) を導入してルール適合な行動を選択
- 出力行動に TTO を適用しさらに補正

### CTG w/o f+g

- 提案手法 (CTG) から ガイダンスと フィルタリングを除いたモデル
- 拡散モデルからランダムにサンプルを出力
- ルールを考慮しない出力との比較用

# 4.2 Single Rule Evaluation

各手法が個別のルールをどれだけ満たしているかを評価・ 走行可能エリアからの逸脱なし (no off-road)

- 速度制限 (speed limit)
- 目標速度 (target speed)
- ゴール到達 (goal waypoint)
- 衝突回避 (no collision)

## 速度制限

- 各シーンの車両速度の上位75%分位値を制限速度として設定  
→ それを超えないように制約

## 目標速度

- 各車両に対して速度の目標値 (正解速度の中央値) を設定  
→ 各ステップでその速度を維持

## ゴール到達

- 15秒後に到達すべき位置を各車両ごとに設定  
→ 正確にそこに到達するよう誘導

## 衝突回避

- 他車両と衝突しないこと

## 逸脱なし

- 道路外に逸脱しないこと

| Rule                         | STL Formula                                                                                                                                                              | Evaluation Metric                                                                                                                       |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| speed limit                  | $\phi_{\text{speedlimit}} := \bigwedge_i \Box (v_{i,t} - v_{\text{limit}_i}) < \epsilon$                                                                                 | $h_{\text{speedlimit}} := \sum_i \sum_t \max(0, v_{i,t} - v_{\text{limit}_i})$                                                          |
| target speed                 | $\phi_{\text{targetspeed}} := \bigwedge_{i,t}  v_{i,t} - v_{\text{target}_{i,t}}  < \epsilon$                                                                            | $h_{\text{targetspeed}} := \sum_i \sum_t  v_{i,t} - v_{\text{target}_{i,t}} $                                                           |
| no collision                 | $\phi_{\text{collision}} := \Box \bigwedge_{i \neq j}   (x_{i,t}, y_{i,t}) - (x_{j,t}, y_{j,t})   > \epsilon$                                                            | $h_{\text{collision}} := \sum_{i \neq j} \mathbb{1}[  (x_{i,t}, y_{i,t}) - (x_{j,t}, y_{j,t})   \leq \epsilon]$                         |
| no off-road                  | $\phi_{\text{offroad}} := \bigwedge_{i,w} \Box   (x_{i,t}, y_{i,t}) - (x_{\text{off-road}_w}, y_{\text{off-road}_w})   > \epsilon$                                       | $h_{\text{offroad}} := \sum_i \mathbb{1}[\min_w   (x_{i,t}, y_{i,t}) - (x_{\text{off-road}_w}, y_{\text{off-road}_w})   \leq \epsilon]$ |
| goal waypoint                | $\phi_{\text{waypoint}} := \bigwedge_i \Diamond   (x_{i,t}, y_{i,t}) - (x_{\text{goal}_i}, y_{\text{goal}_i})   < \epsilon$                                              | $h_{\text{waypoint}} := \sum_i \min_t   (x_{i,t}, y_{i,t}) - (x_{\text{goal}_i}, y_{\text{goal}_i})  $                                  |
| stop sign                    | $\phi_{\text{stopsign}} := \bigwedge_i \Box (\phi_i \text{ in stop region} \Rightarrow \Diamond \Box_{[0,m]} (\phi_i \text{ in stop region} \wedge \phi_{\text{stop}}))$ | $h_{\text{stopsign}} := \sum_i \min_{i \text{ in stop region}}  v_{i,t} $                                                               |
| stop sign + no off-road      | $\phi_{\text{stopsign}} \wedge \phi_{\text{offroad}}$                                                                                                                    | $h_{\text{stopsign}}, h_{\text{offroad}}$                                                                                               |
| goal waypoint + target speed | $\phi_{\text{waypoint}} \wedge \phi_{\text{targetspeed}}$                                                                                                                | $h_{\text{waypoint}}, h_{\text{targetspeed}}$                                                                                           |

# 4.2 Single Rule Evaluation

## 定量的な結果

- CTGは5つの評価設定のうち3つで最も低いルール違反率を達成
- 現実性（realism）と失敗率（fail）の面でもCTGは他手法に匹敵、もしくはそれ以上
- 目標速度（Target Speed）では、CTGが最小違反率 + 現実的な軌跡
- ゴール到達（Goal Waypoint）では、BITS+optが満足度では上回るが、  
→ 曲がりすぎた軌跡により衝突が多発し、現実的ではない

TABLE II: Quantitative results (Single Rule). The top two methods for each metric are highlighted.

|                | speed limit |       |       | target speed |       |       | no collision |       |       | no off-road |       |       | goal waypoint |       |       |
|----------------|-------------|-------|-------|--------------|-------|-------|--------------|-------|-------|-------------|-------|-------|---------------|-------|-------|
|                | rule        | real  | fail  | rule         | real  | fail  | rule         | real  | fail  | rule        | real  | fail  | rule          | real  | fail  |
| SimNet         | 0.739       | 0.898 | 0.353 | 1.989        | 0.898 | 0.353 | 0.137        | 0.898 | 0.353 | 0.453       | 0.900 | 0.353 | 7.543         | 0.900 | 0.353 |
| SimNet+opt     | 0.038       | 0.770 | 0.470 | 0.630        | 1.234 | 0.593 | 0.045        | 1.149 | 0.398 | 0.427       | 1.242 | 0.416 | 1.947         | 1.162 | 0.467 |
| TrafficSim     | 0.737       | 1.362 | 0.443 | 1.922        | 1.346 | 0.440 | 0.140        | 1.542 | 0.416 | 0.485       | 1.564 | 0.401 | 7.733         | 1.574 | 0.458 |
| TrafficSim+opt | 0.042       | 1.444 | 0.325 | 0.610        | 1.634 | 0.404 | 0.075        | 1.063 | 0.265 | 0.423       | 1.836 | 0.413 | 2.414         | 1.766 | 0.532 |
| BITS           | 0.188       | 1.068 | 0.256 | 1.054        | 0.968 | 0.246 | 0.038        | 1.220 | 0.314 | 0.432       | 1.131 | 0.296 | 4.493         | 1.152 | 0.332 |
| BITS+opt       | 0.033       | 1.190 | 0.487 | 0.681        | 1.434 | 0.542 | 0.028        | 1.617 | 0.354 | 0.435       | 1.261 | 0.343 | 1.533         | 1.280 | 0.442 |
| CTG (w/o f+g)  | 1.380       | 0.396 | 0.301 | 2.662        | 0.396 | 0.301 | 0.172        | 0.396 | 0.301 | 0.369       | 0.396 | 0.301 | 7.052         | 0.396 | 0.301 |
| CTG            | 0.019       | 0.359 | 0.165 | 0.150        | 0.855 | 0.179 | 0.040        | 0.569 | 0.271 | 0.341       | 0.501 | 0.455 | 1.943         | 0.564 | 0.387 |

# 4.2 Single Rule Evaluation

## 定性的な結果

### (a)～(c): Target Speed (目標速度)

- CTG は最も滑らかで現実的な軌跡を生成し、違反値も最小 (0.3 m/s)。
- BC+opt, TrafficSim+opt は速度制御がやや不安定。



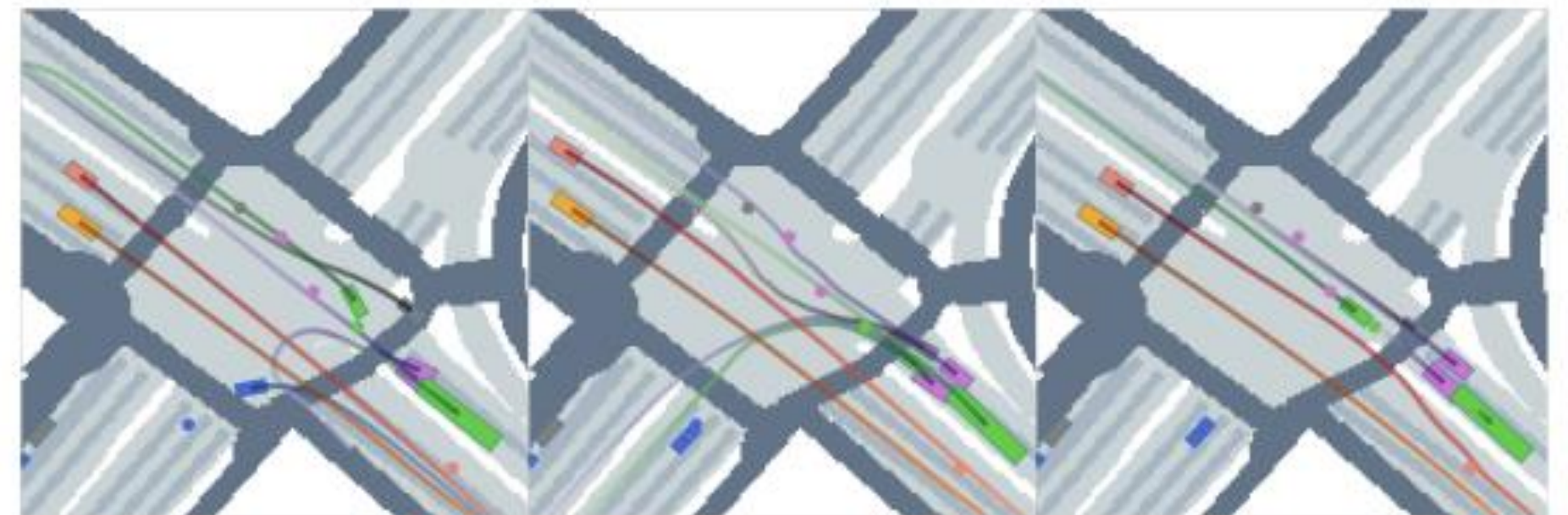
(a) BC+opt(1.3)

(b) TrafficSim+opt(1.1)

(c) CTG(0.3)

### (d)～(f): Goal Waypoint (ゴール到達)

- BC+opt は目標位置への到達度 (ルール満足度) は高いが、軌跡が曲がりすぎて非現実的。衝突も発生。
- BITS+opt も蛇行軌跡で複数の衝突あり。
- CTG はずれはあるが、自然で安定した走行軌跡を維持



(d) BC+opt(2.15)

(e) BITS+opt(6.28)

(f) CTG(3.17)

# 4.3 Multiple Rules Evaluation

## 各手法が複数のルールをどれだけ満たしているかを評価

- 一時停止 + 逸脱なし (stop sign + no off-road)
- ゴール到達 + 目標速度 (goal waypoint + target speed)

### 一時停止 + 逸脱なし

- 一時停止エリア (20×20m) に進入した場合、必ず停止しなければならない  
→ 「停止すること」を時相論理で厳密に定義
- 同時に、道路外へ逸脱しないことも条件に含まれる

2つの制約を同時に満たす必要があり、STLベース手法の性能を測るのに適した設定

### ゴール到達 + 目標速度

- 各車両はGround Truth (正解軌跡) に基づいて設定されたゴールに向かって移動する
- 移動中は毎ステップ、正解速度を再現するように走行

現実的な走行の再現性が求められる設定

| Rule                         | STL Formula                                                                                                                                                                 | Evaluation Metric                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| speed limit                  | $\phi_{\text{speedlimit}} := \bigwedge_i \Box (v_{i,t} - v_{\text{limit}_i}) < \epsilon$                                                                                    | $h_{\text{speedlimit}} := \sum_i \sum_t \max(0, v_{i,t} - v_{\text{limit}_i})$                                                          |
| target speed                 | $\phi_{\text{targetspeed}} := \bigwedge_{i,t}  v_{i,t} - v_{\text{target}_{i,t}}  < \epsilon$                                                                               | $h_{\text{targetspeed}} := \sum_i \sum_t  v_{i,t} - v_{\text{target}_{i,t}} $                                                           |
| no collision                 | $\phi_{\text{collision}} := \Box \bigwedge_{i \neq j} \ (x_{i,t}, y_{i,t}) - (x_{j,t}, y_{j,t})\  > \epsilon$                                                               | $h_{\text{collision}} := \sum_{i \neq j} \mathbb{1}[\ (x_{i,t}, y_{i,t}) - (x_{j,t}, y_{j,t})\  \leq \epsilon]$                         |
| no off-road                  | $\phi_{\text{offroad}} := \bigwedge_{i,w} \Box \ (x_{i,t}, y_{i,t}) - (x_{\text{off-road}_w}, y_{\text{off-road}_w})\  > \epsilon$                                          | $h_{\text{offroad}} := \sum_i \mathbb{1}[\min_w \ (x_{i,t}, y_{i,t}) - (x_{\text{off-road}_w}, y_{\text{off-road}_w})\  \leq \epsilon]$ |
| goal waypoint                | $\phi_{\text{waypoint}} := \bigwedge_i \Diamond \ (x_{i,t}, y_{i,t}) - (x_{\text{goal}_i}, y_{\text{goal}_i})\  < \epsilon$                                                 | $h_{\text{waypoint}} := \sum_i \min_t \ (x_{i,t}, y_{i,t}) - (x_{\text{goal}_i}, y_{\text{goal}_i})\ $                                  |
| stop sign                    | $\phi_{\text{stopsign}} := \bigwedge_i \Box (\phi_i \text{ in stop region} \Rightarrow \Diamond \Box_{[0, m_i]} (\phi_i \text{ in stop region} \wedge \phi_{\text{stop}}))$ | $h_{\text{stopsign}} := \sum_i \min_{i \text{ in stop region}}  v_{i,t} $                                                               |
| stop sign + no off-road      | $\phi_{\text{stopsign}} \wedge \phi_{\text{offroad}}$                                                                                                                       | $h_{\text{stopsign}}, h_{\text{offroad}}$                                                                                               |
| goal waypoint + target speed | $\phi_{\text{waypoint}} \wedge \phi_{\text{targetspeed}}$                                                                                                                   | $h_{\text{waypoint}}, h_{\text{targetspeed}}$                                                                                           |

# 4.3 Multiple Rules Evaluation

## 定量的な結果

- 複数ルール設定（stop sign + no off-road、goal waypoint + target speed）どちらにおいてもCTG系のモデルがルール違反と現実性の両方で最も優れていた
- しかし、失敗率（衝突や逸脱）はやや高めだった

|                | stop sign + no off-road |       |       |       | goal waypoint + target speed |       |       |       |
|----------------|-------------------------|-------|-------|-------|------------------------------|-------|-------|-------|
|                | rule1                   | rule2 | real  | fail  | rule1                        | rule2 | real  | fail  |
| SimNet         | 2.282                   | 0.454 | 0.898 | 0.353 | 3.803                        | 1.610 | 0.898 | 0.353 |
| SimNet+opt     | 1.527                   | 0.480 | 0.796 | 0.443 | 3.238                        | 0.980 | 1.189 | 0.659 |
| TrafficSim     | 2.670                   | 0.484 | 1.605 | 0.409 | 1.583                        | 4.544 | 1.369 | 0.450 |
| TrafficSim+opt | 0.849                   | 0.405 | 1.577 | 0.281 | 2.817                        | 0.995 | 1.398 | 0.529 |
| BITS           | 2.023                   | 0.434 | 1.032 | 0.254 | 2.677                        | 1.077 | 0.919 | 0.286 |
| BITS+opt       | 1.299                   | 0.471 | 1.083 | 0.399 | 3.171                        | 1.019 | 1.455 | 0.601 |
| CTG (w/o f+g)  | 2.573                   | 0.369 | 0.396 | 0.301 | 3.868                        | 2.202 | 0.396 | 0.301 |
| CTG            | 0.528                   | 0.338 | 0.288 | 0.326 | 1.205                        | 0.231 | 0.738 | 0.329 |

# 4.4 Ablation Study.

## ① ガイダンス・フィルタリングの有無

- **gなし**
    - ガイダンスがないと、ルール違反が大幅に増加。
  - **fなし**
    - ガイダンス単体でも効果○。fと組み合わせるより良い。
- **g + f** が最適なペア。

## ② 出力行動最適化（a）の効果

- **aのみ**
  - Guidanceの代わりにaを使用 → 全指標で悪化。
- **g + a**
  - ルール違反は減少するが、realismが崩れ、失敗率も増加。

## ③ 最適化ステップ数の違い（op）

- **op=3やop=5**  
最適化ステップ数を増やすとルール違反はさらに減少するが、不自然な動きになり現実性や安定性が損なわれる。

**dyn** : 車両運動方程式（一輪車モデル）に基づいた制約を課す。  
**f** : フィルタリング。複数サンプルからルール適合度が高いものを選ぶ  
**g** : ガイダンス。STLガイド関数の勾配を使い軌跡をルール準拠に補正  
**a** : 最終出力行動をさらにルール準拠に近づけるための追加最適化。  
**op** : Guidance中に行う最適化の繰り返し回数。  
**action-only** : 状態ではなく、行動だけを学習目標とする。

|      | Features |   |   |   |    | Loss | Metrics       |               |               |
|------|----------|---|---|---|----|------|---------------|---------------|---------------|
|      | dyn      | f | g | a | op |      | rule          | real          | fail          |
| 提案手法 | ✓        | ✓ | ✓ |   | 1  |      | 0.0189        | 0.3588        | <b>0.1647</b> |
| ①    | ✓        |   |   |   | 1  |      | 1.3802        | 0.3964        | 0.3011        |
|      | ✓        |   | ✓ |   | 1  |      | 0.0387        | 0.4336        | 0.2227        |
|      | ✓        | ✓ |   |   | 1  |      | 1.3268        | 0.3869        | 0.2942        |
| ②    | ✓        |   | ✓ | ✓ | 1  |      | 0.019         | 0.447         | 0.1879        |
|      | ✓        | ✓ | ✓ | ✓ | 1  |      | <b>0.0132</b> | 0.4431        | 0.1845        |
|      | ✓        | ✓ |   | ✓ | 1  |      | 0.0329        | 0.4591        | 0.1881        |
|      | ✓        |   |   | ✓ | 1  |      | 0.0819        | 0.4339        | 0.2117        |
| ③    | ✓        | ✓ | ✓ |   | 3  |      | 0.0158        | 0.4591        | 0.2017        |
|      | ✓        | ✓ | ✓ |   | 5  |      | 0.0156        | 0.456         | 0.1966        |
| ④    | ✓        | ✓ | ✓ |   | 1  | ✓    | 0.018         | <b>0.2057</b> | 0.2824        |
|      |          | ✓ | ✓ |   | 1  |      | 0.0525        | 3.4204        | 0.612         |
|      |          |   |   |   | 1  |      | 1.2797        | 3.3251        | 0.7523        |

# 4.4 Ablation Study.

## ④ 行動のみを教師信号とした場合 (action-only)

- 加速度が滑らかで、軌跡の現実性 (realism) は高いが、失敗率上昇

## ⑤ 物理モデルを無効に (dynなし)

- すべての指標で性能が大きく劣化。  
→ダイナミクス制約 (運動モデル) は必須。

最良構成 (提案手法)

Guidance (g) + Filtration (f) + 物理制約 (dyn) + 状態・行動の教師信号

**dyn** : 車両運動方程式 (一輪車モデル) に基づいた制約を課す。  
**f** : フィルタリング。複数サンプルからルール適合度が高いものを選ぶ  
**g** : ガイダンス。STLガイド関数の勾配を使い軌跡をルール準拠に補正  
**a** : 最終出力行動をさらにルール準拠に近づけるための追加最適化。  
**op** : Guidance中に行う最適化の繰り返し回数。  
**action-only** : 状態ではなく、行動だけを学習目標とする。

|      | Features |   |   |   |    | Loss | Metrics       |               |               |
|------|----------|---|---|---|----|------|---------------|---------------|---------------|
|      | dyn      | f | g | a | op |      | rule          | real          | fail          |
| 提案手法 | ✓        | ✓ | ✓ |   | 1  |      | 0.0189        | 0.3588        | <b>0.1647</b> |
| ①    | ✓        |   |   |   | 1  |      | 1.3802        | 0.3964        | 0.3011        |
|      | ✓        |   | ✓ |   | 1  |      | 0.0387        | 0.4336        | 0.2227        |
|      | ✓        | ✓ |   |   | 1  |      | 1.3268        | 0.3869        | 0.2942        |
| ②    | ✓        |   | ✓ | ✓ | 1  |      | 0.019         | 0.447         | 0.1879        |
|      | ✓        | ✓ | ✓ | ✓ | 1  |      | <b>0.0132</b> | 0.4431        | 0.1845        |
|      | ✓        | ✓ |   | ✓ | 1  |      | 0.0329        | 0.4591        | 0.1881        |
|      | ✓        |   |   | ✓ | 1  |      | 0.0819        | 0.4339        | 0.2117        |
| ③    | ✓        | ✓ | ✓ |   | 3  |      | 0.0158        | 0.4591        | 0.2017        |
|      | ✓        | ✓ | ✓ |   | 5  |      | 0.0156        | 0.456         | 0.1966        |
| ④    | ✓        | ✓ | ✓ |   | 1  | ✓    | 0.018         | <b>0.2057</b> | 0.2824        |
|      |          | ✓ | ✓ |   | 1  |      | 0.0525        | 3.4204        | 0.612         |
|      |          |   |   |   | 1  |      | 1.2797        | 3.3251        | 0.7523        |

# 5. Conclusion

# 6. Conclusions

## 研究目的

- 交通シミュレーションに対して拡散モデルベースの手法 CTG を提案。

## CTGの特徴

- 条件付き拡散モデルにより「制御可能な」軌跡を生成。
- ルールガイド（STL）を通じて、安全性や現実性を制約。

## 現時点での適用範囲・課題

- 車両のみに適用。歩行者や自転車などのマルチエージェント拡張が今後の課題。

## 応用可能性

- 衝突や逸脱のルールにより、より長期的・頑健なシミュレーションが可能。
- 自動運転以外のタスク（例: ロボティクス、意思決定システム）にも適用できる。
- オンラインで動作し、逐次的に反応するような応用にも期待。

# 所感

- 昨年も拡散モデルの論文を読んだが、当時は画像生成への応用しかイメージできず、軌跡生成への応用を理解できてよかった
- 論文自体が短く、細かい説明が少なかったため、理解が難しい部分もあった。
- 明示的にルールをかけて、それぞれ評価できるところが良いと思った
- 車だけでなく、人の軌跡にも使えそう
- この論文ではルール制約は2つまでだったが、それ以上増やすとどうなるのか。。