平均速度場モデルを用いた単一画像からの3次元点群生成

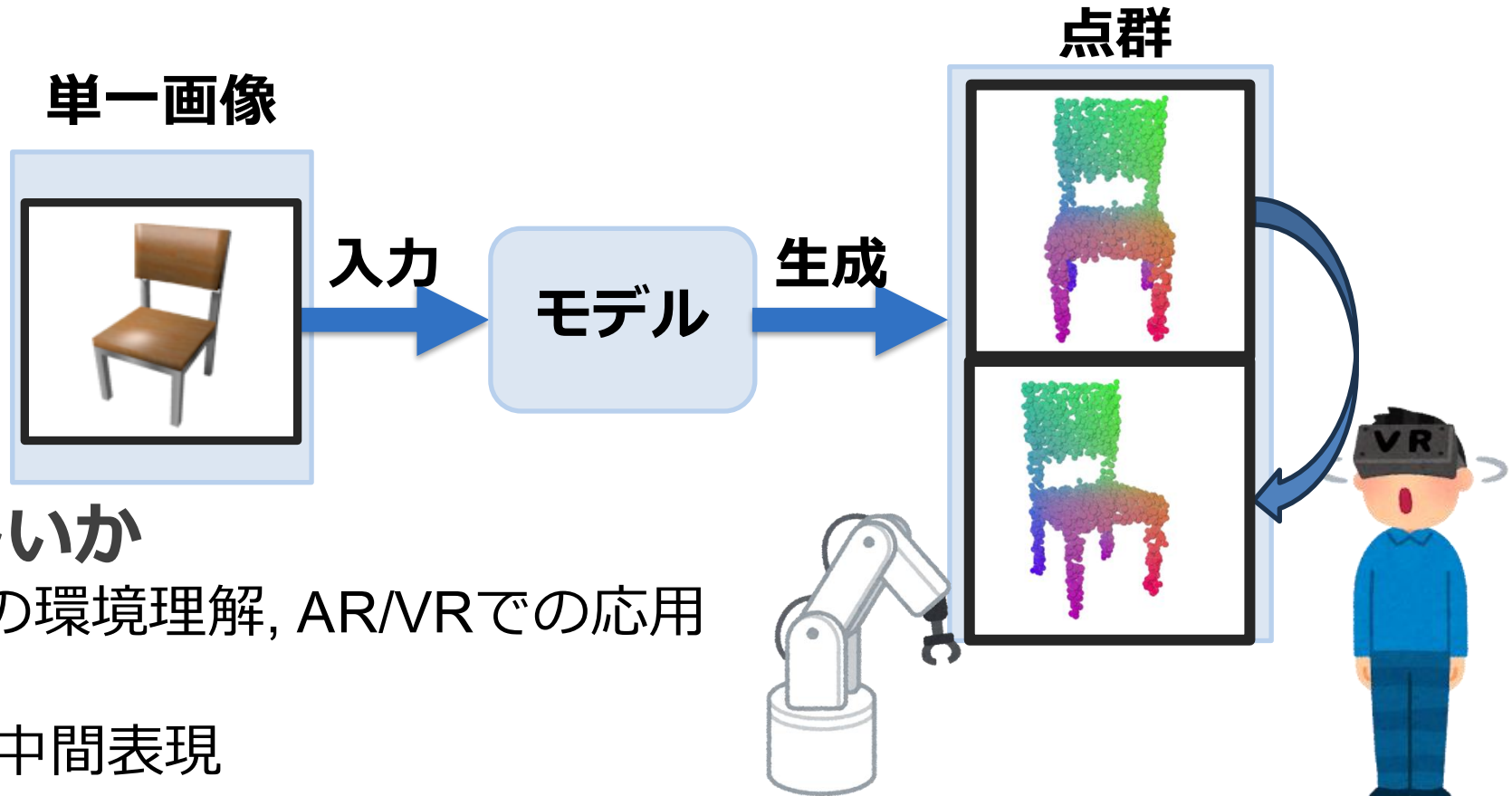
電気通信大学

馬場雄大

Yuta Baba

背景

- **単一画像3次元点群生成とは？**
  - 1枚の画像に映り込んだ物体の**完全な点群**を生成するタスク



- **何がうれしいか**
  - ロボットの環境理解, AR/VRでの応用
  - 再構成の中間表現

課題

- **単一画像⇒3Dは不良設定 (見えない領域が定まらない)**

既存の生成方法

- フィードフォワード型モデル：**高速**だが, 見えない領域の表現が弱い
- 拡散モデルベース：**高品質**だが, 推論で多回数の反復が必要

(例) 拡散モデル [2]の生成

本研究の焦点

- **少ない反復 (低NFE) で高速に拡散モデルを動かす**
  - NFE = Network Function Evaluations (ネットワーク評価回数)

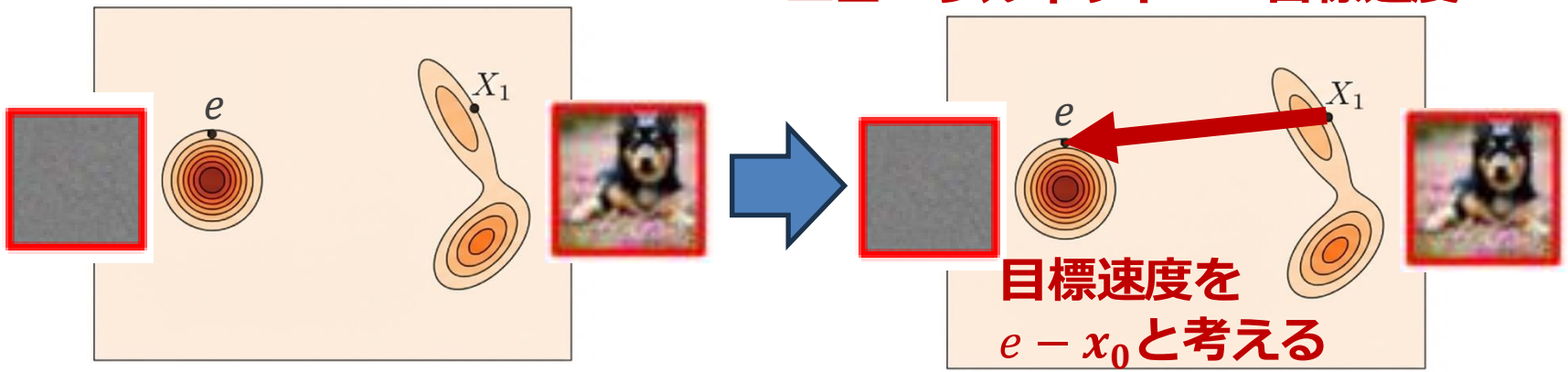
[2] Jonathan Ho et al. "Denoising diffusion probabilistic models." NeurIPS, 2020.

関連研究：深層生成モデル

- **Flow Matching [3]：ノイズからデータ空間への速度場を学習**
  - 速度場：時刻tと現在位置x_tに依存する速度
  - 基本損失：

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_t\sim p_t}\left[\left\| v_t(x_t;\theta) - u_t(x_t)\right\|_2^2\right]$$

ニューラルネット　　目標速度



  - 学習損失：

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\left\| v_t(x_t;\theta) - (e - x_0)\right\|_2^2\right]$$

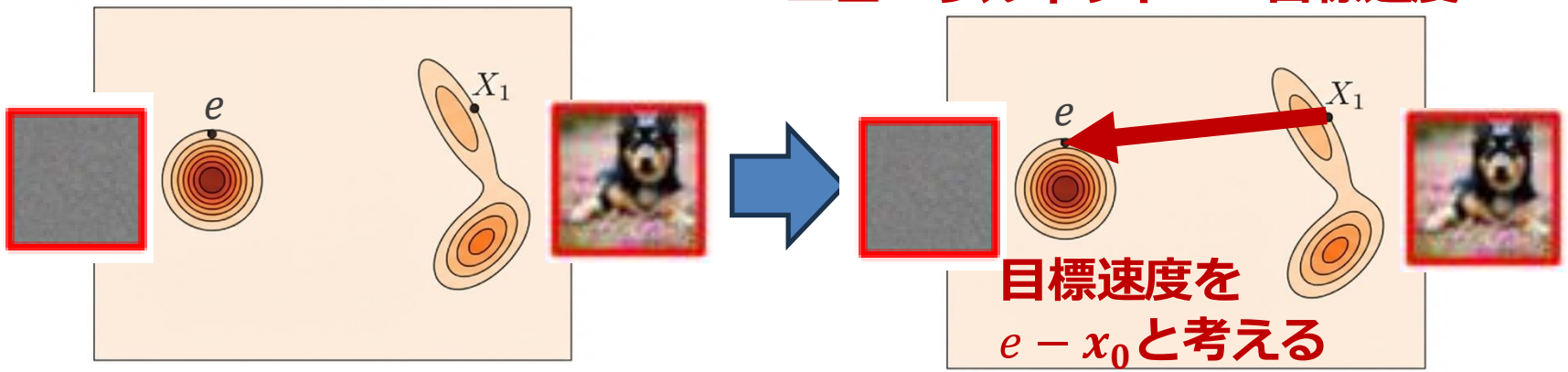
[3] Yaron Lipman et al. "Flow matching for generative modeling." ICLR, 2023.

関連研究：深層生成モデル

- **Flow Matching [3]：ノイズからデータ空間への速度場を学習**
  - 速度場：時刻tと現在位置x_tに依存する速度
  - 基本損失：

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_t\sim p_t}\left[\left\|v_t(x_t;\theta) - u_t(x_t)\right\|_2^2\right]$$

ニューラルネット　　目標速度



  - 学習損失：

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\left\|v_t(x_t;\theta) - (e - x_0)\right\|_2^2\right]$$

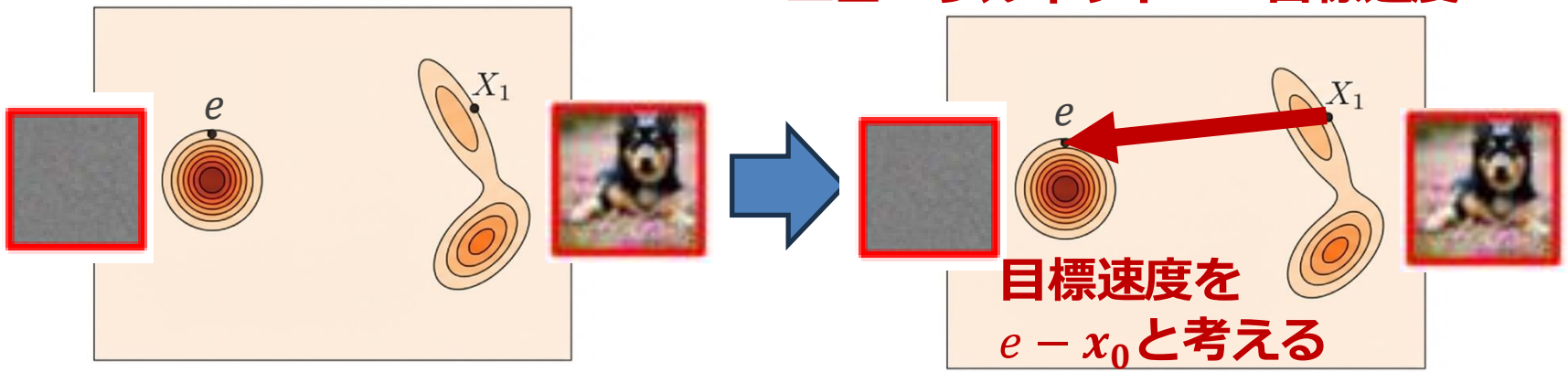
[3] Yaron Lipman et al. "Flow matching for generative modeling." ICLR, 2023.

関連研究：深層生成モデル

- **Flow Matching [3]：ノイズからデータ空間への速度場を学習**
  - 速度場：時刻tと現在位置x_tに依存する速度
  - 基本損失：

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\, x_t\sim p_t}\left[\left\| v_t(x_t;\theta) - u_t(x_t)\right\|_2^2\right]$$

ニューラルネット　　目標速度



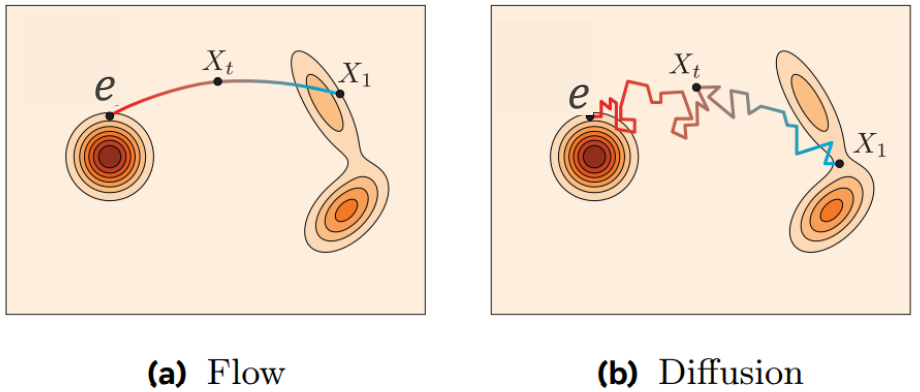
  - 学習損失：

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\left\| v_t(x_t;\theta) - (e - x_0)\right\|_2^2\right]$$

[3] Yaron Lipman et al. "Flow matching for generative modeling." ICLR, 2023.

関連研究：深層生成モデル

❑ Flow Matchingの問題点



(a) Flow　(b) Diffusion

図1：Flow Matchingと拡散モデルの違い

- ➢ 拡散モデルよりもNFEを削減できるが…
  - ▷ 実際の速度場は歪んでしまうので, 極端に減らせない

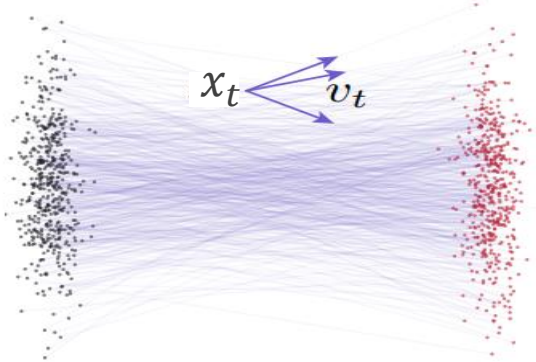


図2：Flow Matchingが学習した速度場

❑ Mean Flow [4]

- ➢ Flow Matchingを**瞬間速度v**と捉え, 平均uを取る
- ➢ 基本式：

$$u(z_t, r, t) = \frac{1}{t-r}\int_r^t v(z_\tau, \tau)\, d\tau$$

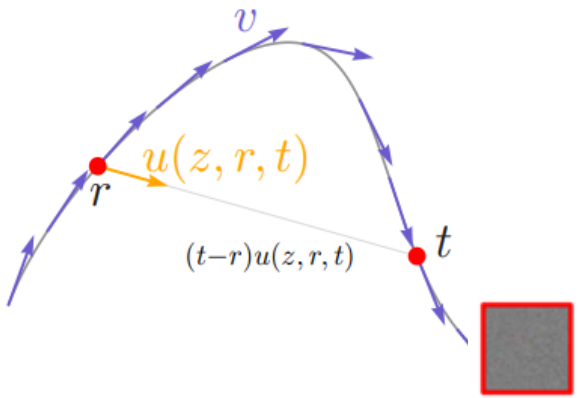


図3：平均速度場

[4] Zhengyang Geng et al. "Mean flows for one-step generative modeling." NeurIPS, 2025.

関連研究：深層生成モデル

❑ Flow Matchingの問題点

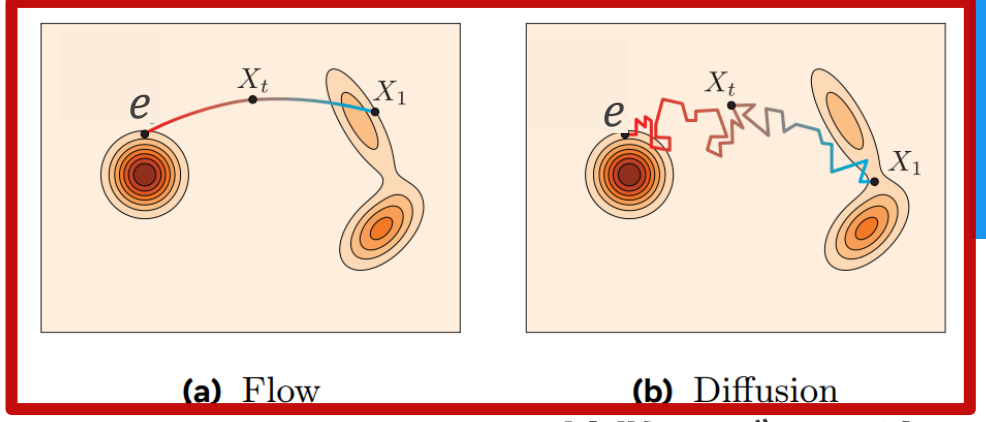


図1：Flow Matchingと拡散モデルの違い

- ➢ 拡散モデルよりもNFEを削減できるが…
  - ▷ 実際の速度場は歪んでしまうので, 極端に減らせない

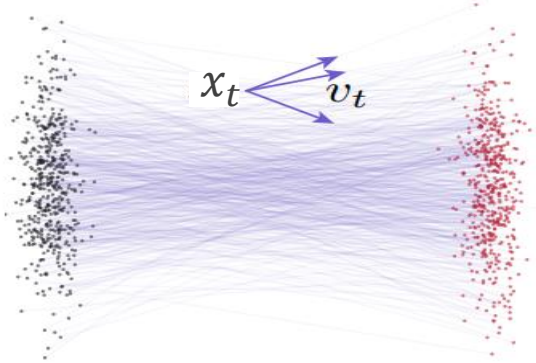


図2：Flow Matchingが学習した速度場

❑ Mean Flow [4]

- ➢ Flow Matchingを**瞬間速度v**と捉え, 平均uを取る
- ➢ 基本式：

$$u(z_t, r, t) = \frac{1}{t-r}\int_r^t v(z_\tau, \tau)\, d\tau$$

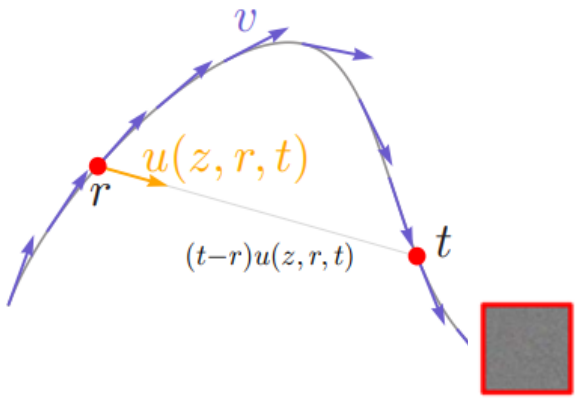


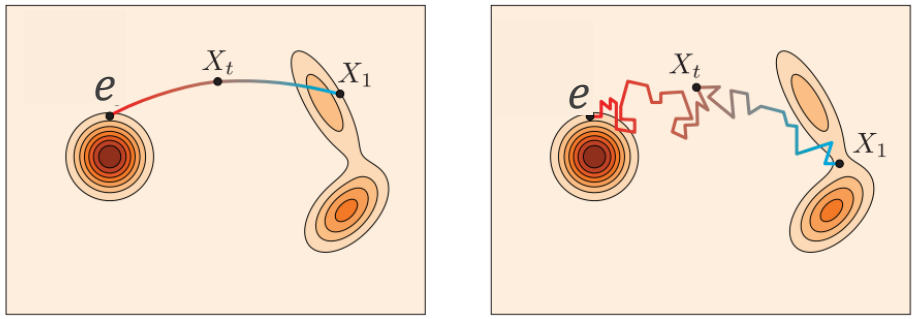
図3：平均速度場

[4] Zhengyang Geng et al. "Mean flows for one-step generative modeling." NeurIPS, 2025.

関連研究：深層生成モデル

❑ Flow Matchingの問題点

- ➢ 拡散モデルよりもNFEを削減できるが…
  - ▷ 実際の速度場は歪んでしまうので, 極端に減らせない



(a) Flow (b) Diffusion

図1：Flow Matchingと拡散モデルの違い

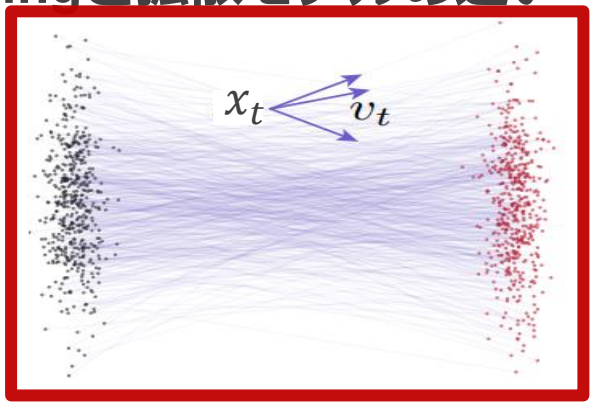


図2：Flow Matchingが学習した速度場

❑ Mean Flow [4]

- ➢ Flow Matchingを**瞬間速度v**と捉え, 平均uを取る
- ➢ 基本式：

$$u(z_t, r, t) = \frac{1}{t-r}\int_r^t v(z_\tau, \tau)\, d\tau$$

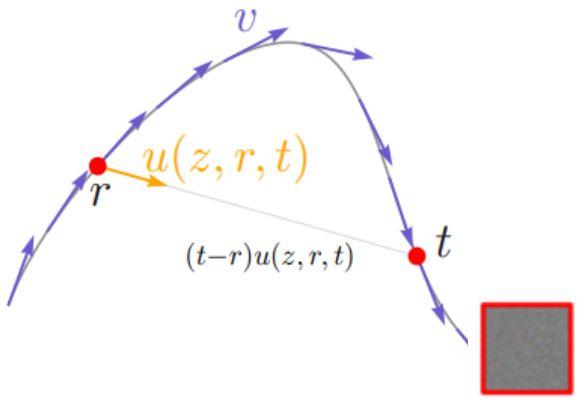
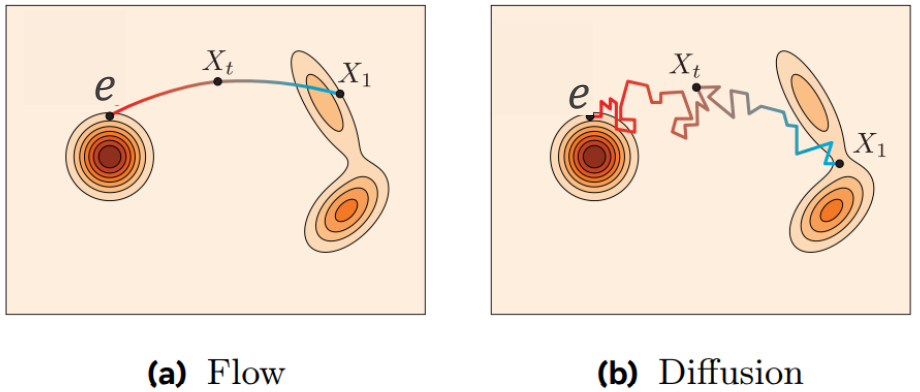


図3：平均速度場

[4] Zhengyang Geng et al. "Mean flows for one-step generative modeling." NeurIPS, 2025.

関連研究：深層生成モデル

❑ Flow Matchingの問題点



(a) Flow　(b) Diffusion

図1：Flow Matchingと拡散モデルの違い

- ➢ 拡散モデルよりもNFEを削減できるが…
  - ▷ 実際の速度場は歪んでしまうので, 極端に減らせない

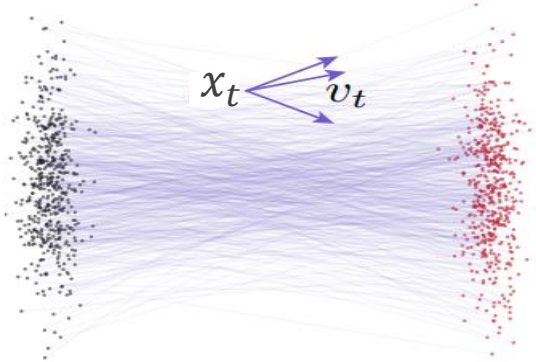


図2：Flow Matchingが学習した速度場

❑ Mean Flow [4]

- ➢ Flow Matchingを**瞬間速度v**と捉え, 平均uを取る
- ➢ 基本式：

$$u(z_t, r, t) = \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, d\tau$$

瞬間速度を積分

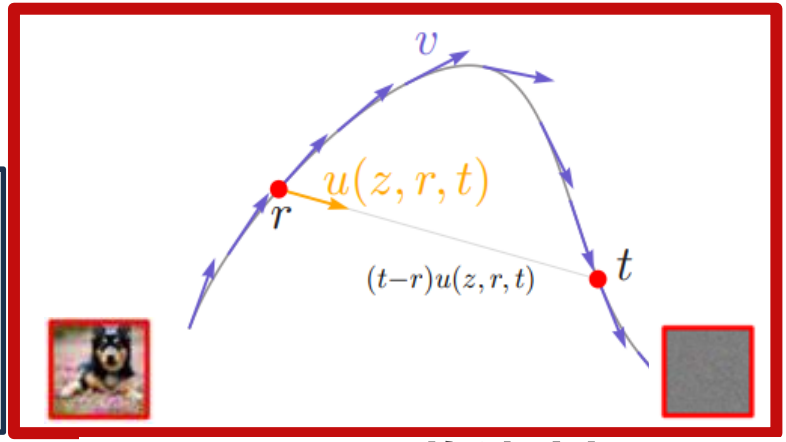


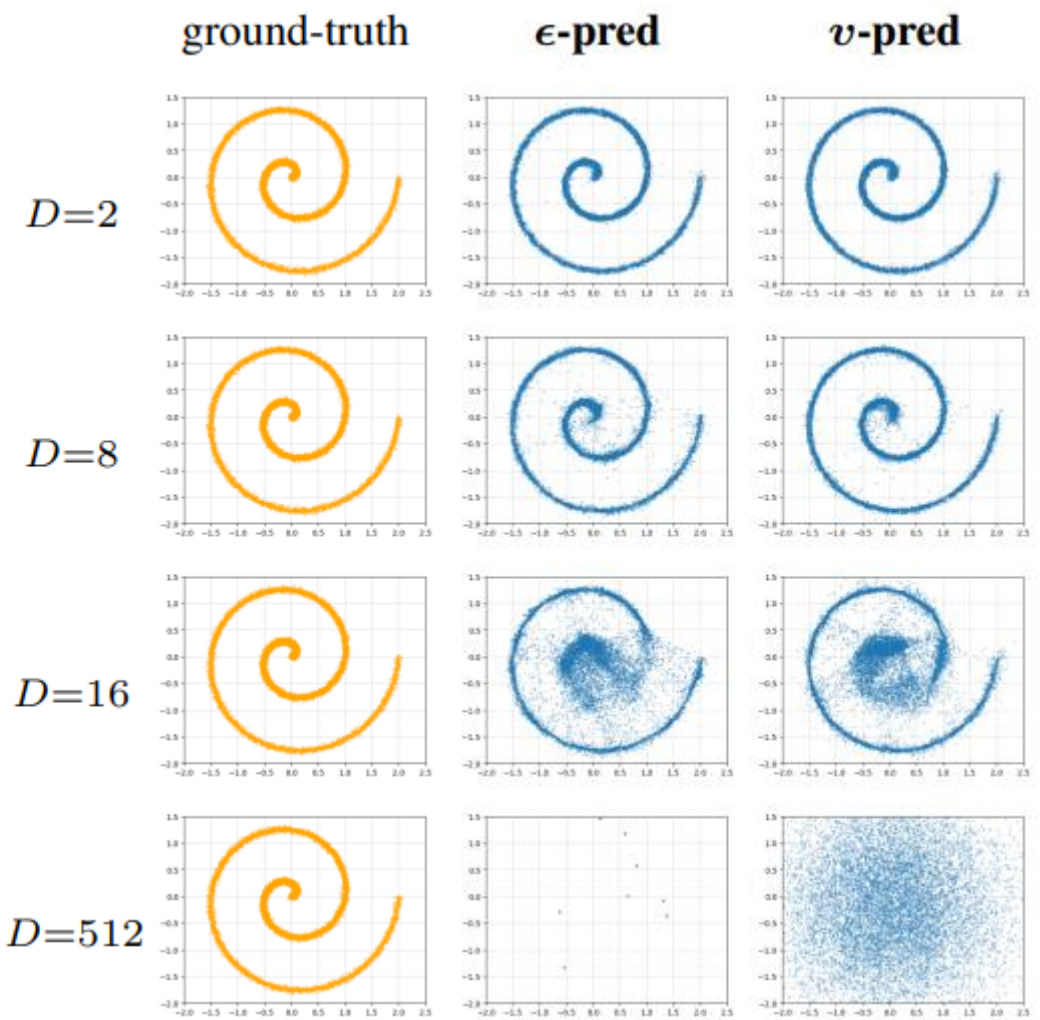
図3：平均速度場

[4] Zhengyang Geng et al. "Mean flows for one-step generative modeling." NeurIPS, 2025.

関連研究：深層生成モデル

❑ Mean Flowの問題点

- ➢ 主な実証がVAEを用いた**潜在空間**のみ
  - ▷ **速度場**による実データ空間の予測は高次元で悪化 [8]
  - ▷ 点群では, VAEによるエンコード, デコードは複雑
    - 三次元畳み込み, k近傍探索等…

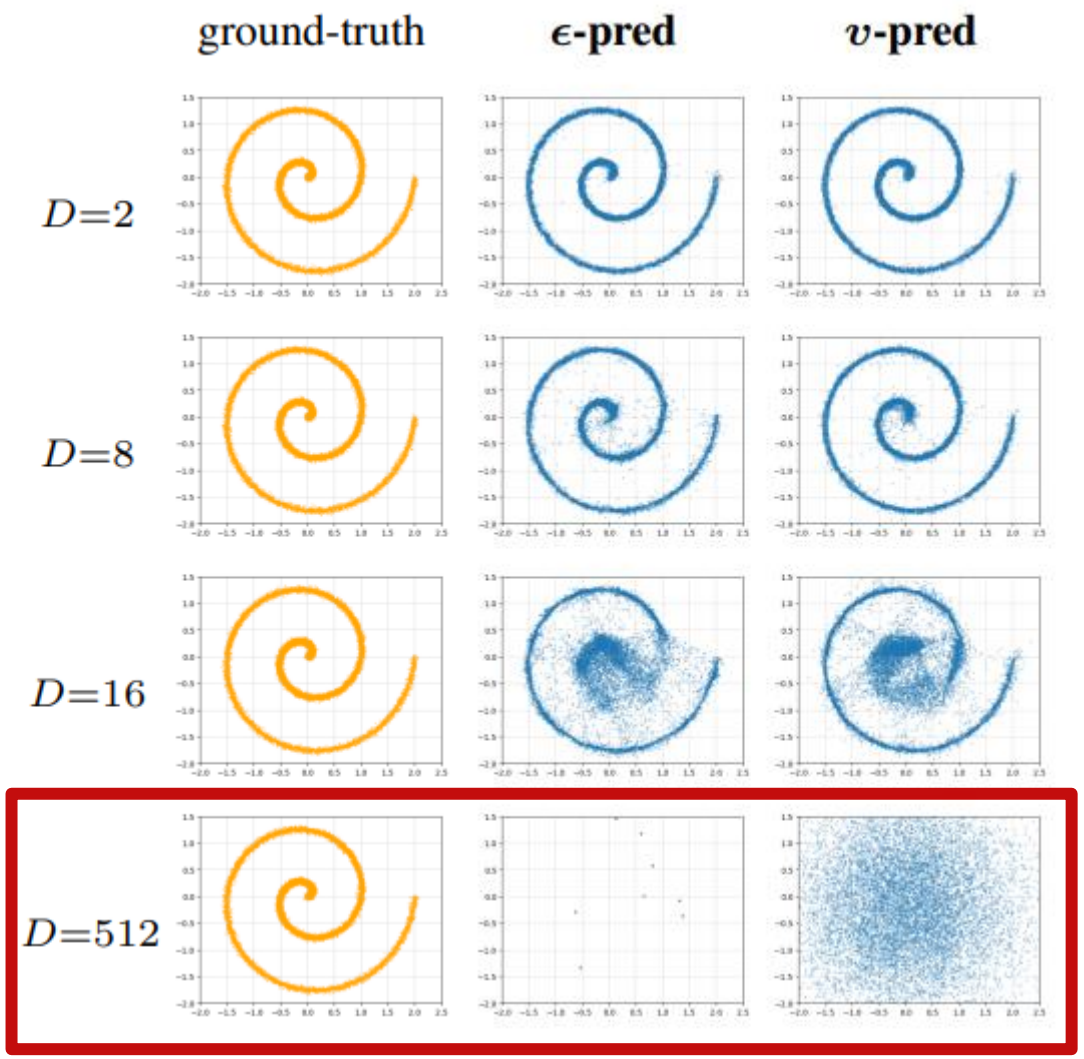


[8] Tianhong Li and Kaiming He. "Back to basics: Let denoising generative models denoise." arXiv, 2025.

関連研究：深層生成モデル

❑ Mean Flowの問題点

- ➢ 主な実証がVAEを用いた**潜在空間**のみ
  - ▷ **速度場**による実データ空間の予測は高次元で悪化 [8]
  - ▷ 点群では, VAEによるエンコード, デコードは複雑
    - 三次元畳み込み, k近傍探索等…

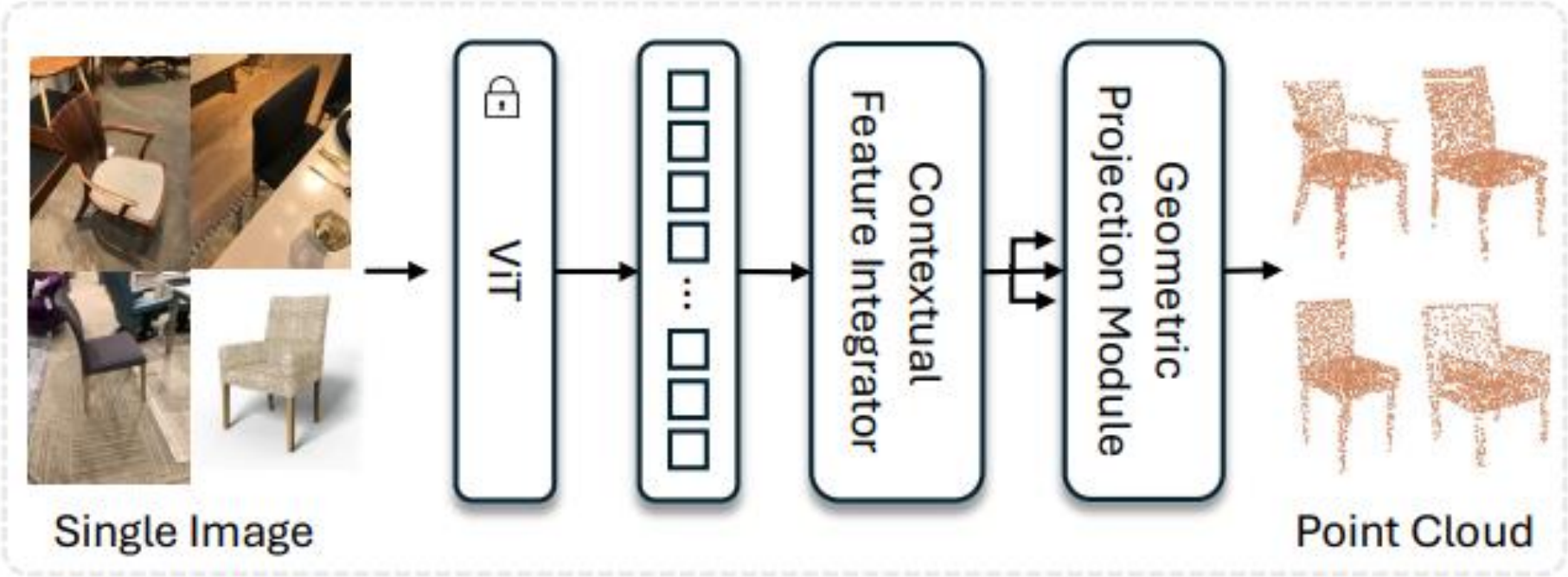


[8] Tianhong Li and Kaiming He. "Back to basics: Let denoising generative models denoise." arXiv, 2025.

関連研究：単一画像点群再構成モデル

▫ フィードフォワード型モデル

- ➢ RGB2point [13]
  - ▷ 凍結したVITのトークンを, 後段の点群予測器でChamfer Distance損失

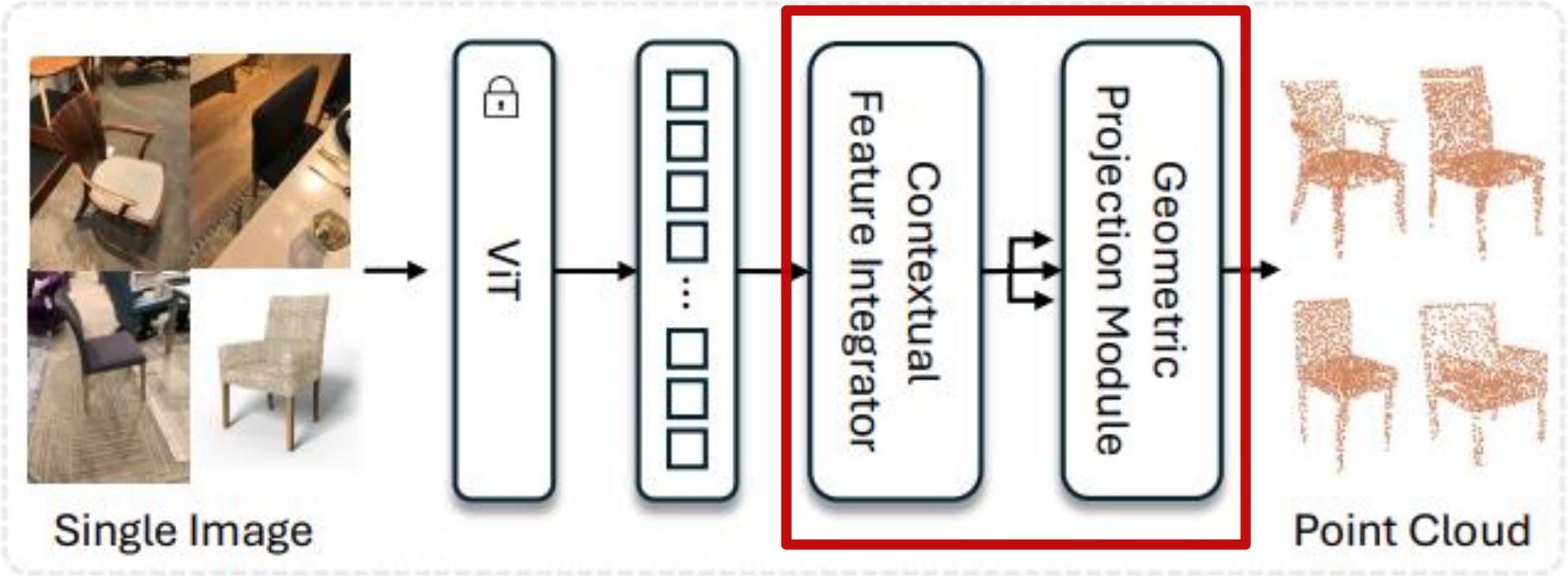


[13] J. J. Lee and B. Benes. "Rgb2point: 3D point cloud generation from single RGB images." WACV, 2025.

関連研究：単一画像点群再構成モデル

❑ フィードフォワード型モデル

- ➢ RGB2point [13]
  - ▷ 凍結したVITのトークンを, 後段の点群予測器でChamfer Distance損失

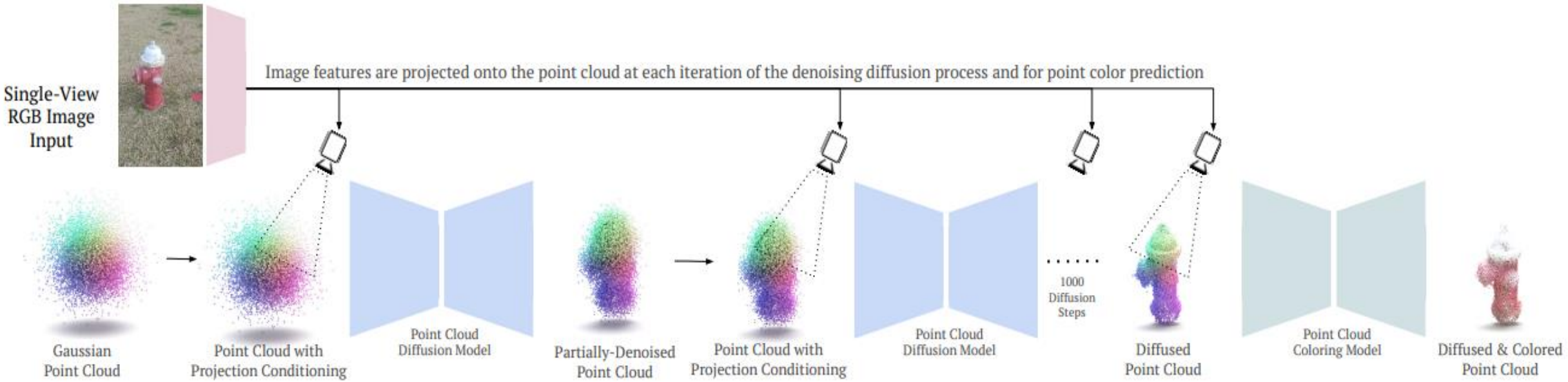


[13] J. J. Lee and B. Benes. "Rgb2point: 3D point cloud generation from single RGB images." WACV, 2025.

関連研究：単一画像点群再構成モデル

❑ 拡散モデルベース

- ➢ PC^2 [22]
  - ▷ 拡散モデルの条件付けを，明示的なカメラパラメータとともに行う．

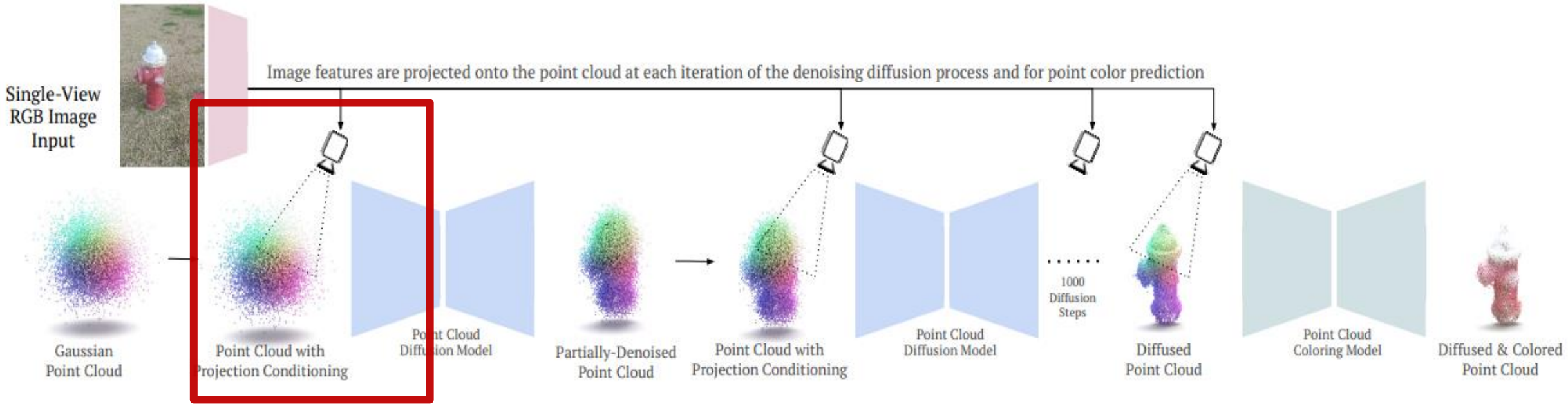


[22] L. Melas-Kyriazi, C. Rupprecht, and A. Vedaldi. "PC^2: Projection-conditioned point cloud diffusion for single-image 3D reconstruction." CVPR, 2023.

関連研究：単一画像点群再構成モデル

❑ 拡散モデルベース

- ➢ PC^2 [22]
  - ▷ 拡散モデルの条件付けを，明示的なカメラパラメータとともに行う．

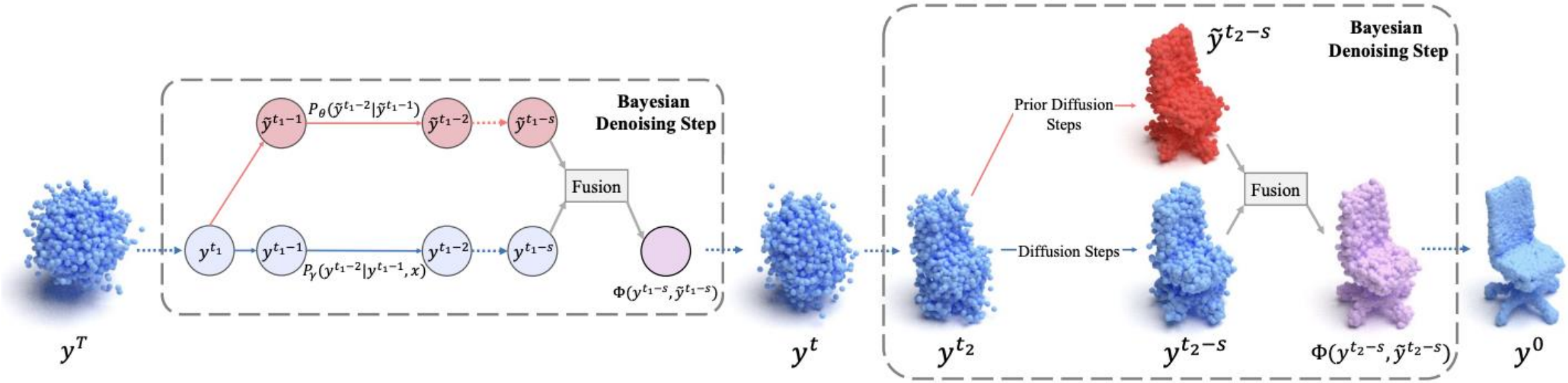


[22] L. Melas-Kyriazi, C. Rupprecht, and A. Vedaldi. "PC^2: Projection-conditioned point cloud diffusion for single-image 3D reconstruction." CVPR, 2023.

関連研究：単一画像点群再構成モデル

❑ 拡散モデルベース

- ➢ Bayesian Diffusion Models (BDM) [51]
  - ▷ 3D形状の事前知識 (prior) モデルと, 画像から復元モデルの融合

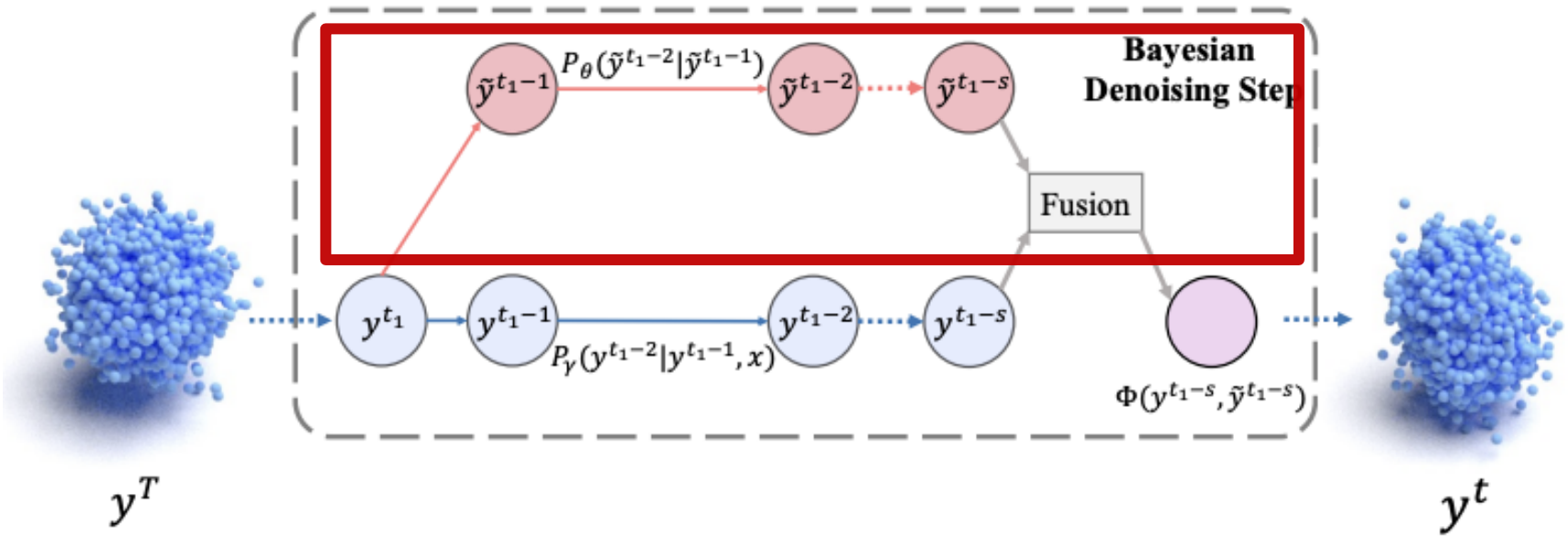


[51] H. Xu, Y. Lei, Z. Chen, X. Zhang, Y. Zhao, Y. Wang, and Z. Tu. "Bayesian diffusion models for 3D shape reconstruction." CVPR, 2024.

関連研究：単一画像点群再構成モデル

❑ 拡散モデルベース

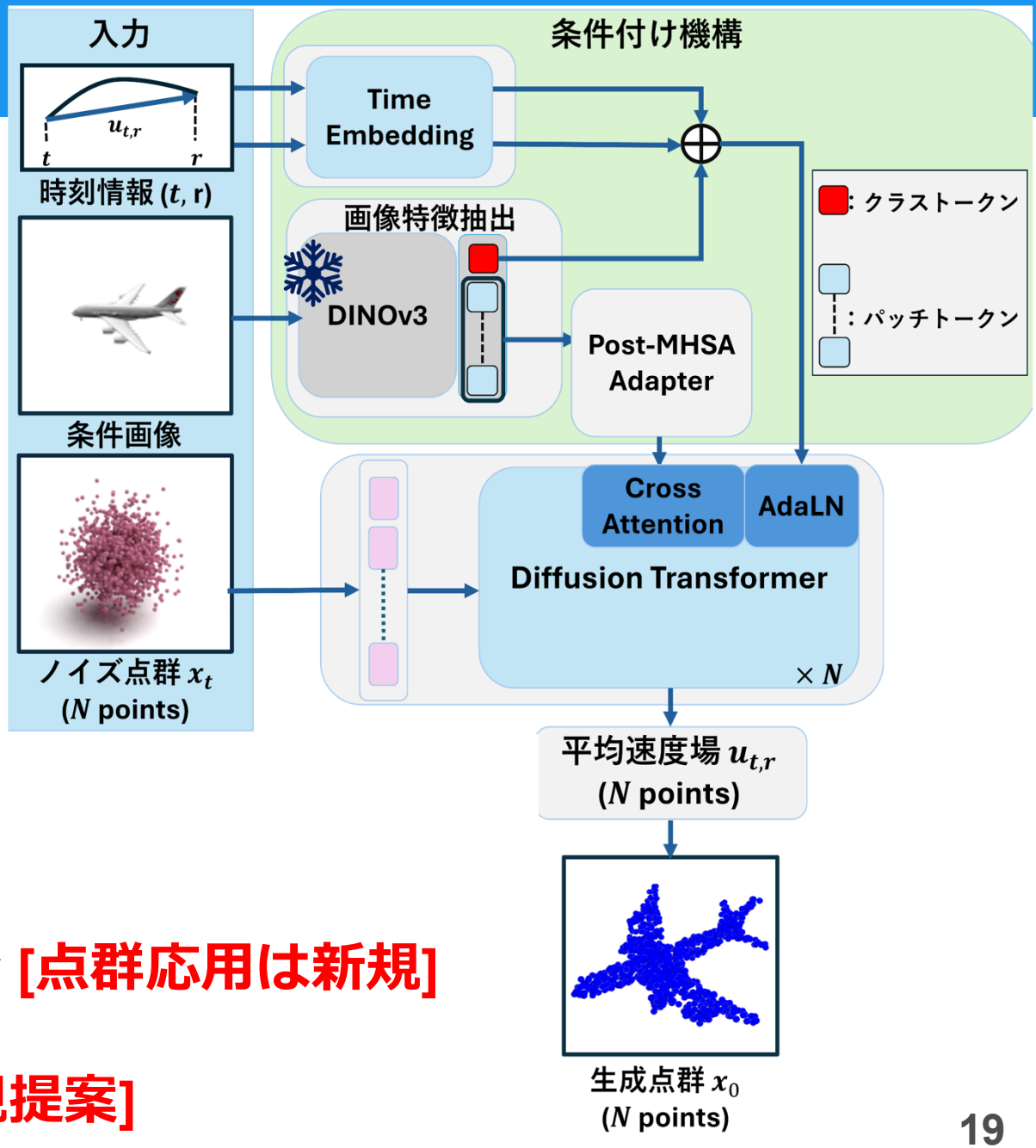
- ➢ Bayesian Diffusion Models (BDM) [51]
  - ▷ 3D形状の事前知識 (prior) モデルと, 画像から復元モデルの融合



[51] H. Xu, Y. Lei, Z. Chen, X. Zhang, Y. Zhao, Y. Wang, and Z. Tu. "Bayesian diffusion models for 3D shape reconstruction." CVPR, 2024.

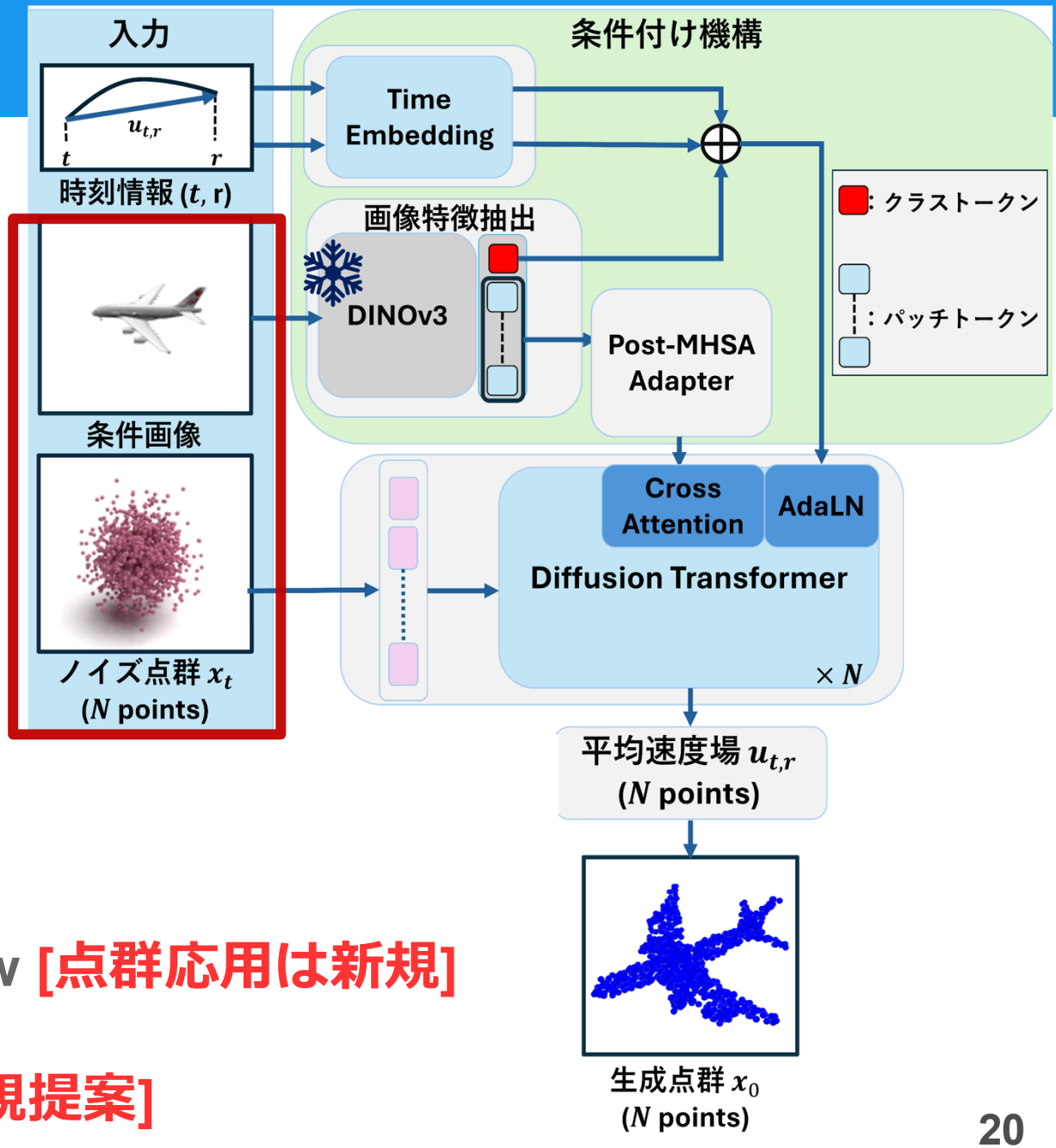
提案手法：全体像

- 生成の流れ
  - ノイズ点群と画像を入力
  - 画像を条件に点群を生成
- 目次
  - アーキテクチャ
    - Diffusion Transformer (DiT)
      - Post-MHSA Adapter [新規提案]
  - 損失
    - Classifier Free Guidance Mean Flow [点群応用は新規]
    - Geometry Noise Anchor (GNA) [新規提案]



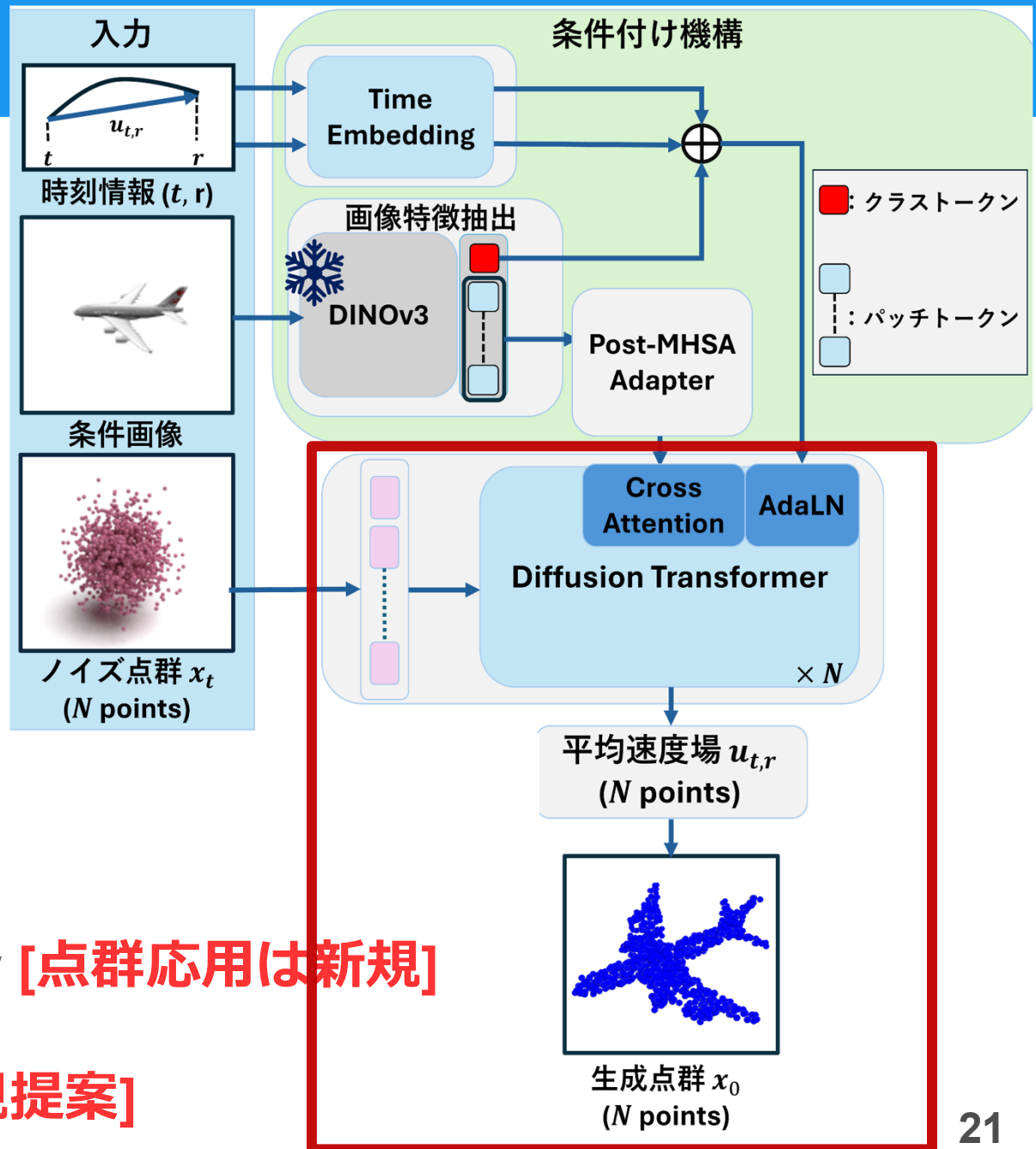
提案手法：全体像

- 生成の流れ
  - ノイズ点群と画像を入力
  - 画像を条件に点群を生成
- 目次
  - アーキテクチャ
    - Diffusion Transformer (DiT)
      - Post-MHSA Adapter [新規提案]
  - 損失
    - Classifier Free Guidance Mean Flow [点群応用は新規]
    - Geometry Noise Anchor (GNA) [新規提案]



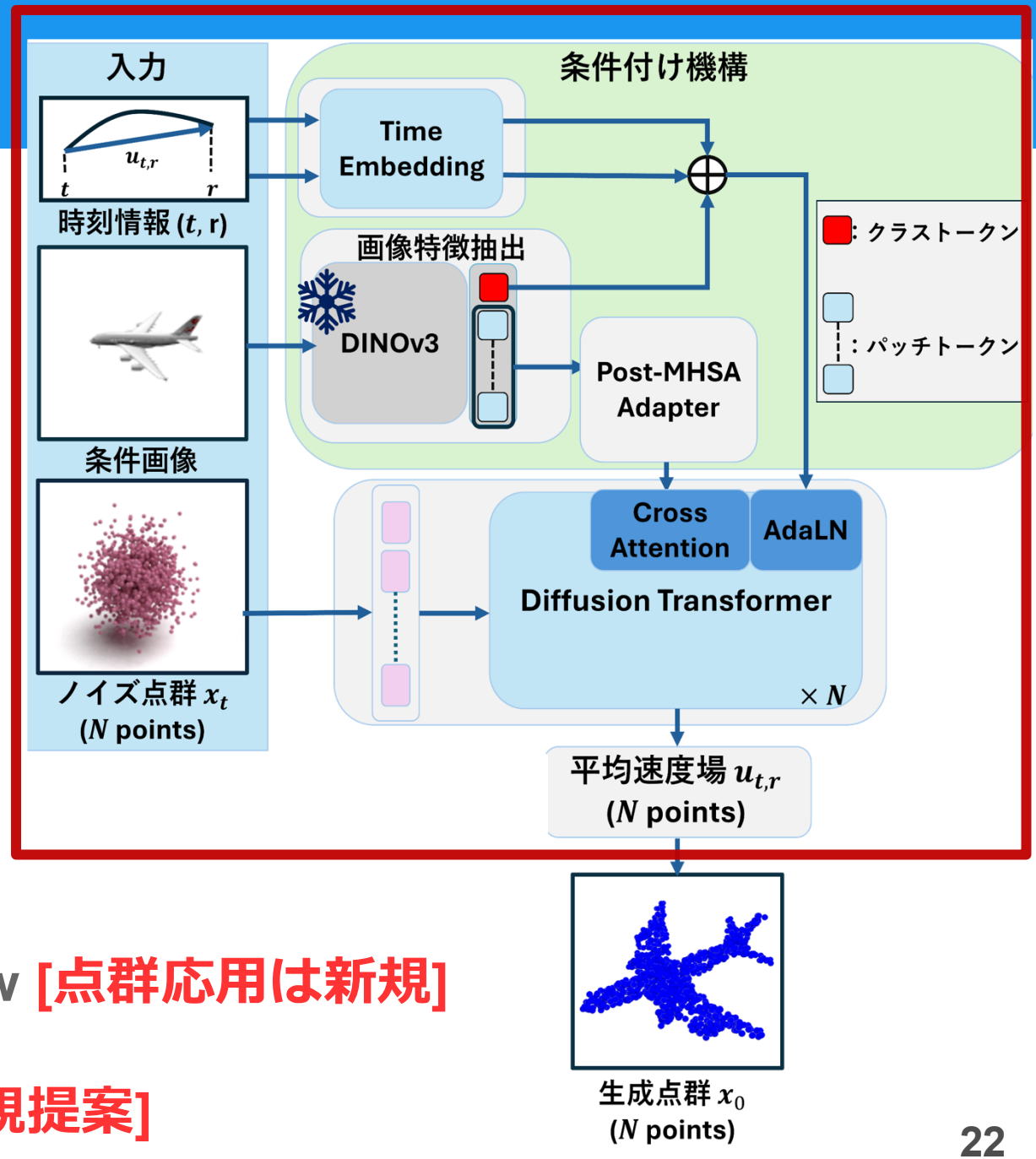
提案手法：全体像

- 生成の流れ
  - ノイズ点群と画像を入力
  - **画像を条件に点群を生成**
- 目次
  - アーキテクチャ
    - Diffusion Transformer (DiT)
      - Post-MHSA Adapter **[新規提案]**
  - 損失
    - Classifier Free Guidance Mean Flow **[点群応用は新規]**
    - Geometry Noise Anchor (GNA) **[新規提案]**



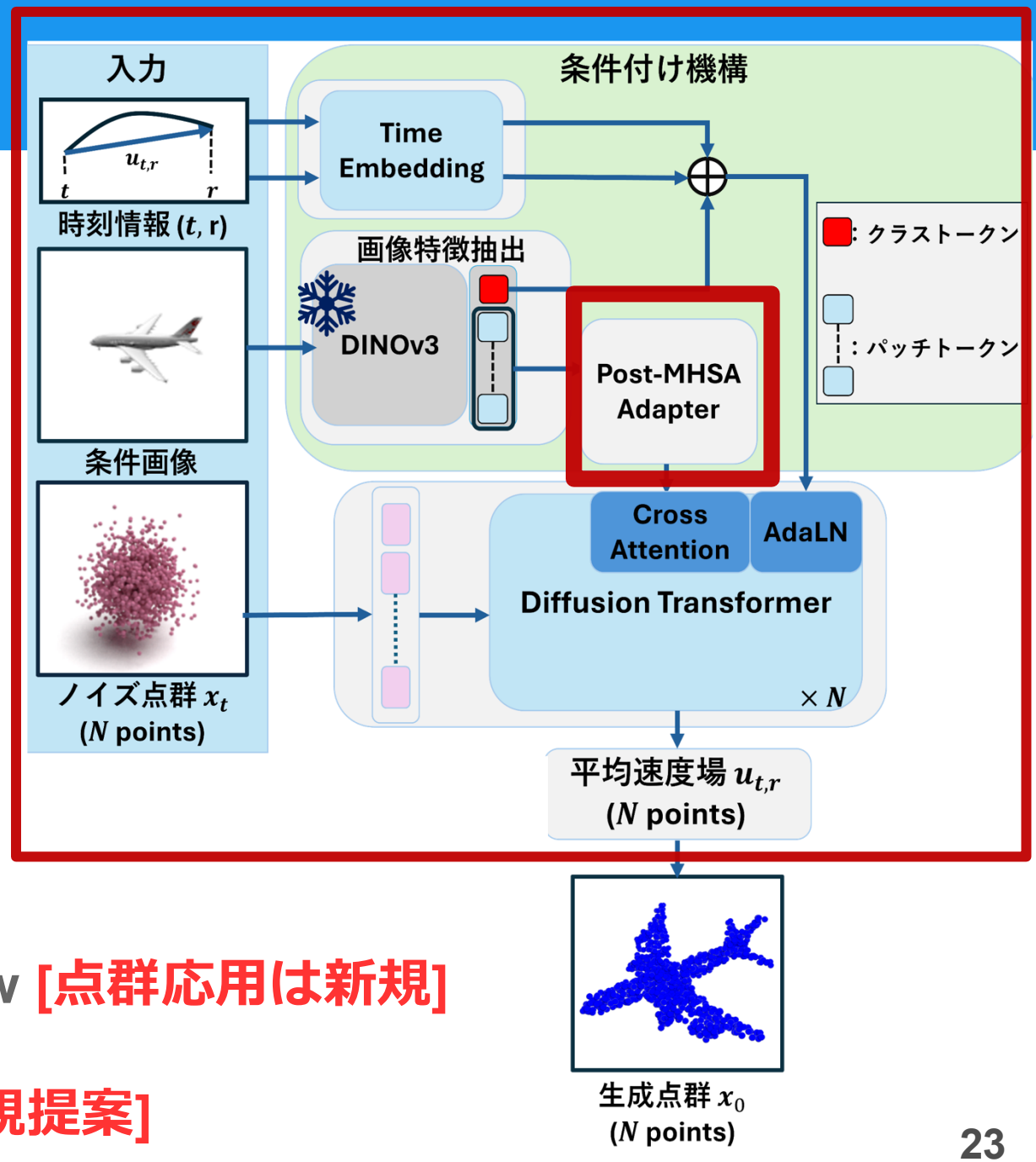
提案手法：全体像

- 生成の流れ
  - ノイズ点群と画像を入力
  - 画像を条件に点群を生成
- 目次
  - アーキテクチャ
    - Diffusion Transformer (DiT)
      - Post-MHSA Adapter [新規提案]
  - 損失
    - Classifier Free Guidance Mean Flow [点群応用は新規]
    - Geometry Noise Anchor (GNA) [新規提案]



提案手法：全体像

- 生成の流れ
  - ノイズ点群と画像を入力
  - 画像を条件に点群を生成
- 目次
  - アーキテクチャ
    - Diffusion Transformer (DiT)
      - Post-MHSA Adapter [新規提案]
  - 損失
    - Classifier Free Guidance Mean Flow [点群応用は新規]
    - Geometry Noise Anchor (GNA) [新規提案]



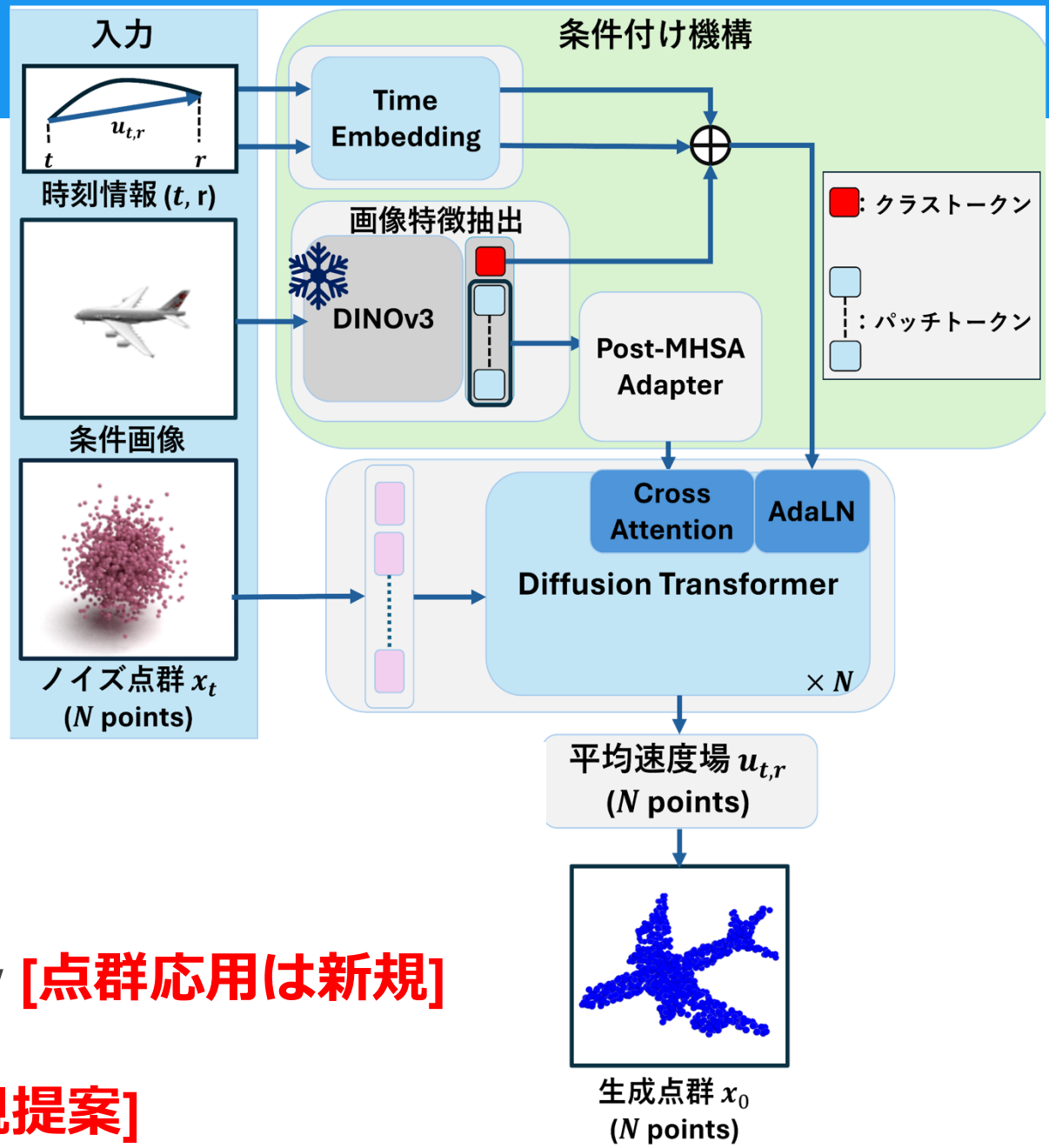
提案手法：全体像

❑ 生成の流れ

- ➢ ノイズ点群と画像を入力
- ➢ 画像を条件に点群を生成

❑ 目次

- ➢ アーキテクチャ
  - ▷ Diffusion Transformer (DiT)
    - • Post-MHSA Adapter [新規提案]
- ➢ 損失
  - ▷ Classifier Free Guidance Mean Flow [点群応用は新規]
  - ▷ Geometry Noise Anchor (GNA) [新規提案]

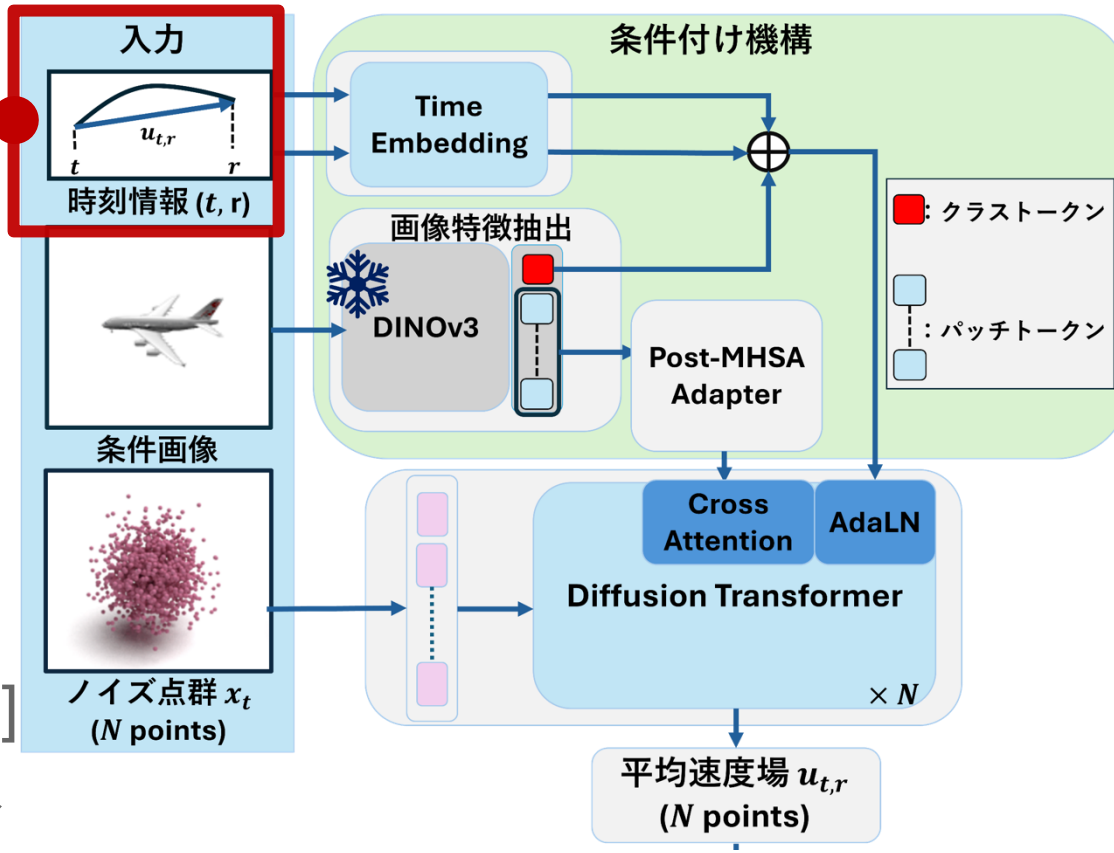


提案手法：Diffusion Transformer

❑ Diffusion Transformer

➢ Mean Flowは**現在時刻t**と**次時刻r**が必要

$$u(z_t, r, t) = \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, d\tau$$



▷ AdaLN-Zero [16] で**各時刻**とDINOv3 [18] で得られた画像の**クラス特徴**の和を注入

▷ Cross Attentionで**画像パッチ特徴**を入れる

[16] William Peebles and Saining Xie. "Scalable diffusion models with transformers." ICCV, 2023.
[18] Oriane Simeoni et al. "Dinov3." arXiv:2508.10104, 2025.

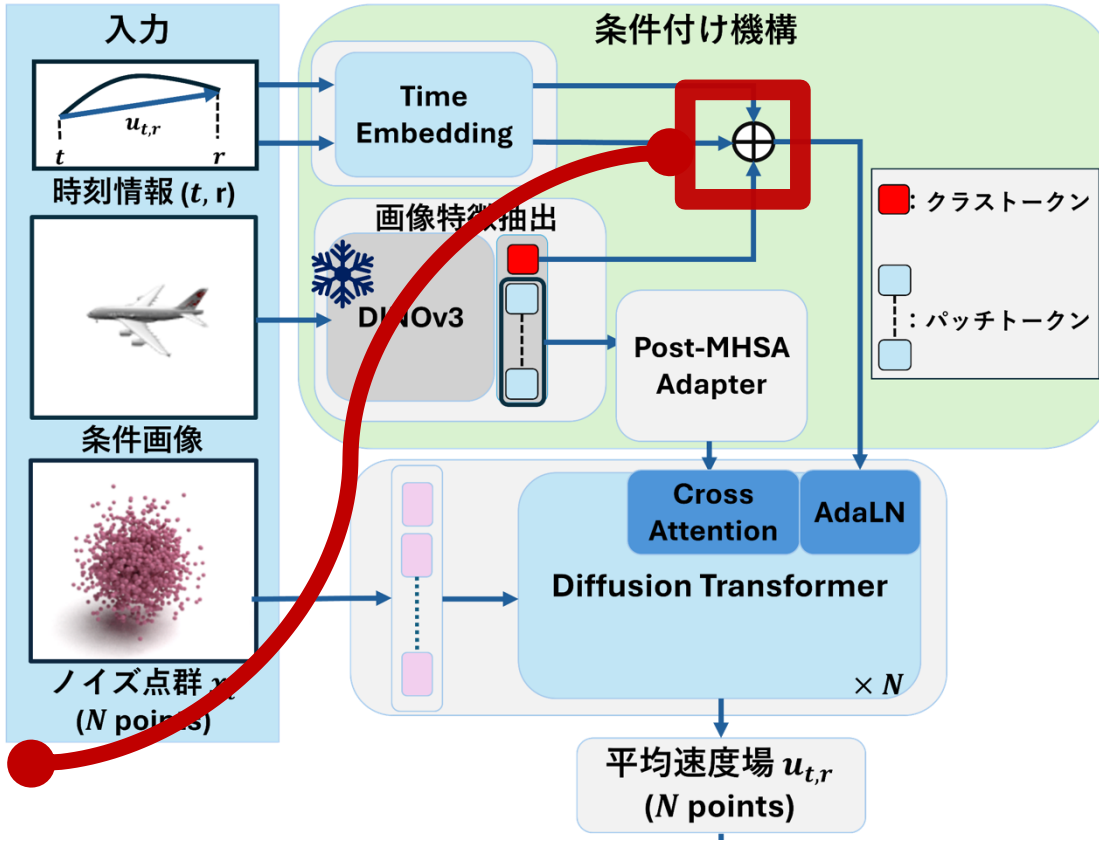
提案手法：Diffusion Transformer

❑ Diffusion Transformer

➢ Mean Flowは**現在時刻t**と**次時刻r**が必要

$$u(z_t, r, t) = \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, d\tau$$

▷ AdaLN-Zero [16] で**各時刻**とDINOv3 [18] で得られた画像の**クラス特徴**の和を注入



▷ Cross Attentionで**画像パッチ特徴**を入れる

[16] William Peebles and Saining Xie. "Scalable diffusion models with transformers." ICCV, 2023.
[18] Oriane Simeoni et al. "Dinov3." arXiv:2508.10104, 2025.

提案手法：Diffusion Transformer

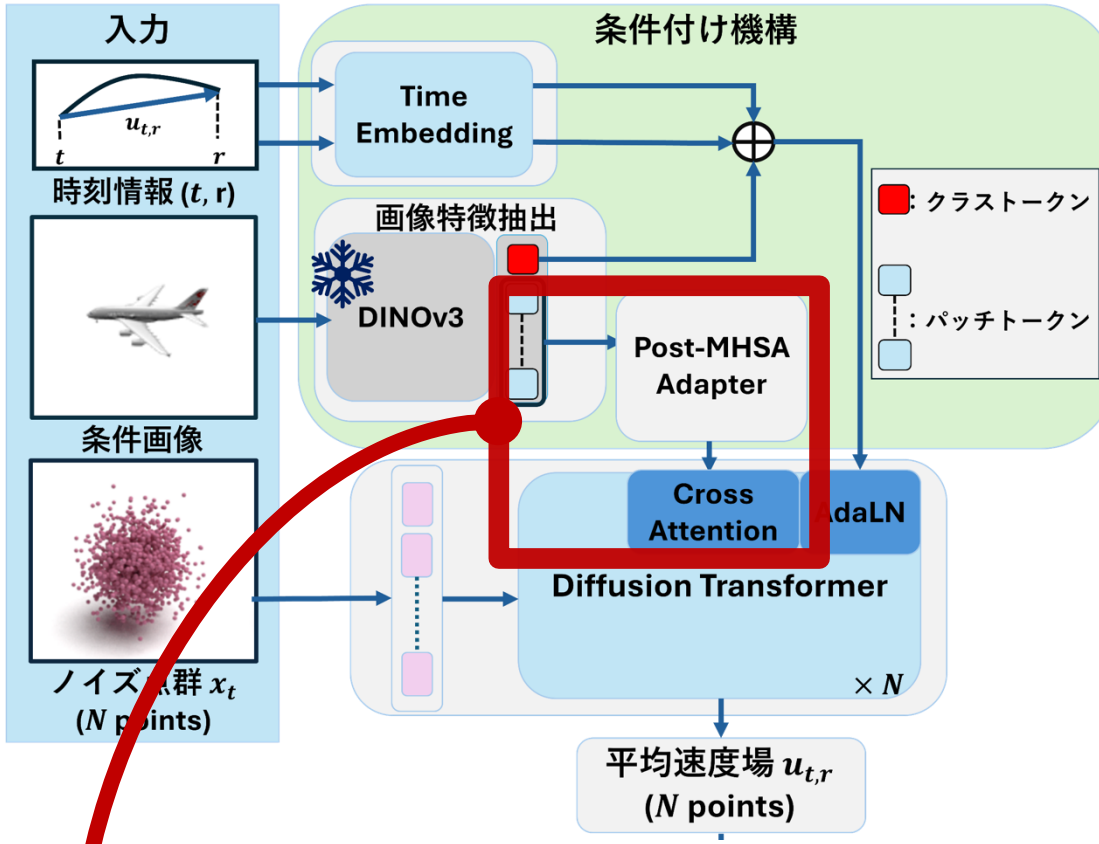
❑ Diffusion Transformer

➢ Mean Flowは**現在時刻t**と**次時刻r**が必要

$$u(z_t, r, t) = \frac{1}{t-r}\int_r^t v(z_\tau, \tau)\, d\tau$$

▷ AdaLN-Zero [16] で**各時刻**とDINOv3 [18] で得られた画像の**クラス特徴**の和を注入

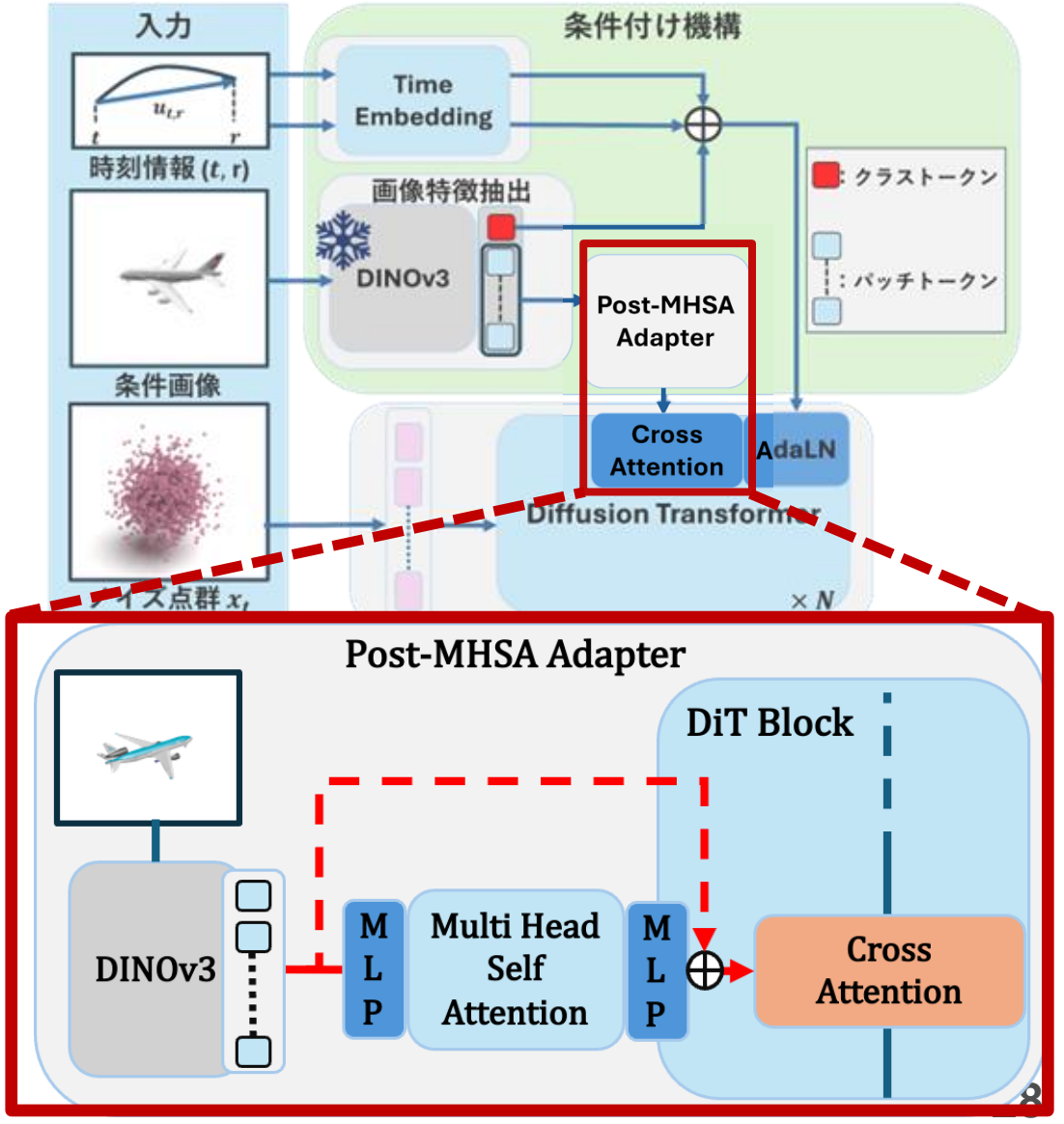
▷ Cross Attentionで**画像パッチ特徴**を入れる



[16] William Peebles and Saining Xie. "Scalable diffusion models with transformers." ICCV, 2023.
[18] Oriane Simeoni et al. "Dinov3." arXiv:2508.10104, 2025.

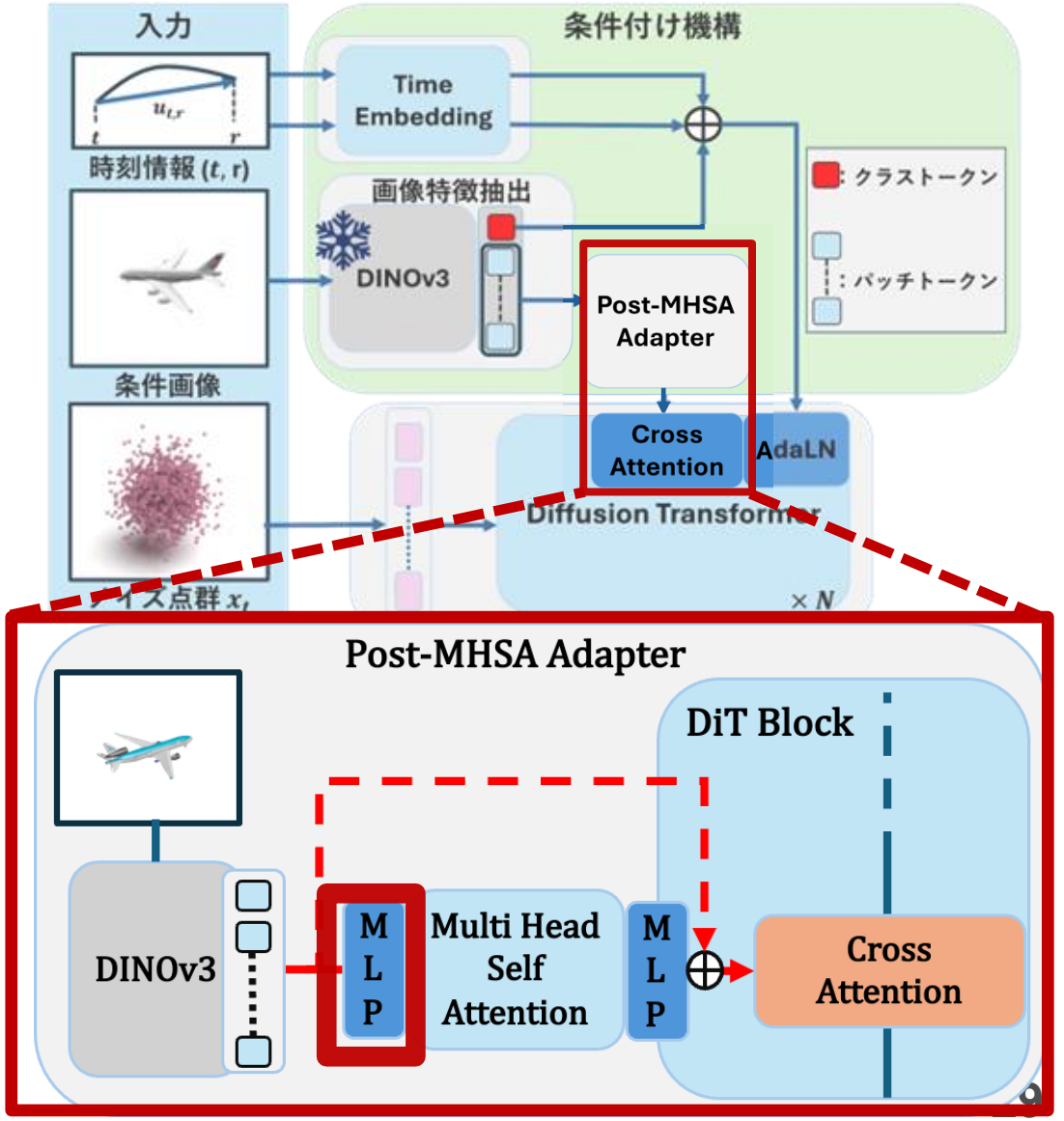
提案手法：Post-MHSA Adapter [新規提案]

- Cross Attentionによる細部情報の入力
  - ➢ パッチ特徴は画像細部の特徴を持つが…
    - ▷ 点群特徴に最適化されていない可能性
    - ▷ 背景などから不要な情報が入る可能性
      - Self Attentionで重要領域を計算
      - DiTの各層のMLPでタスク適応
      - 元の特徴を保つため, 残差接続



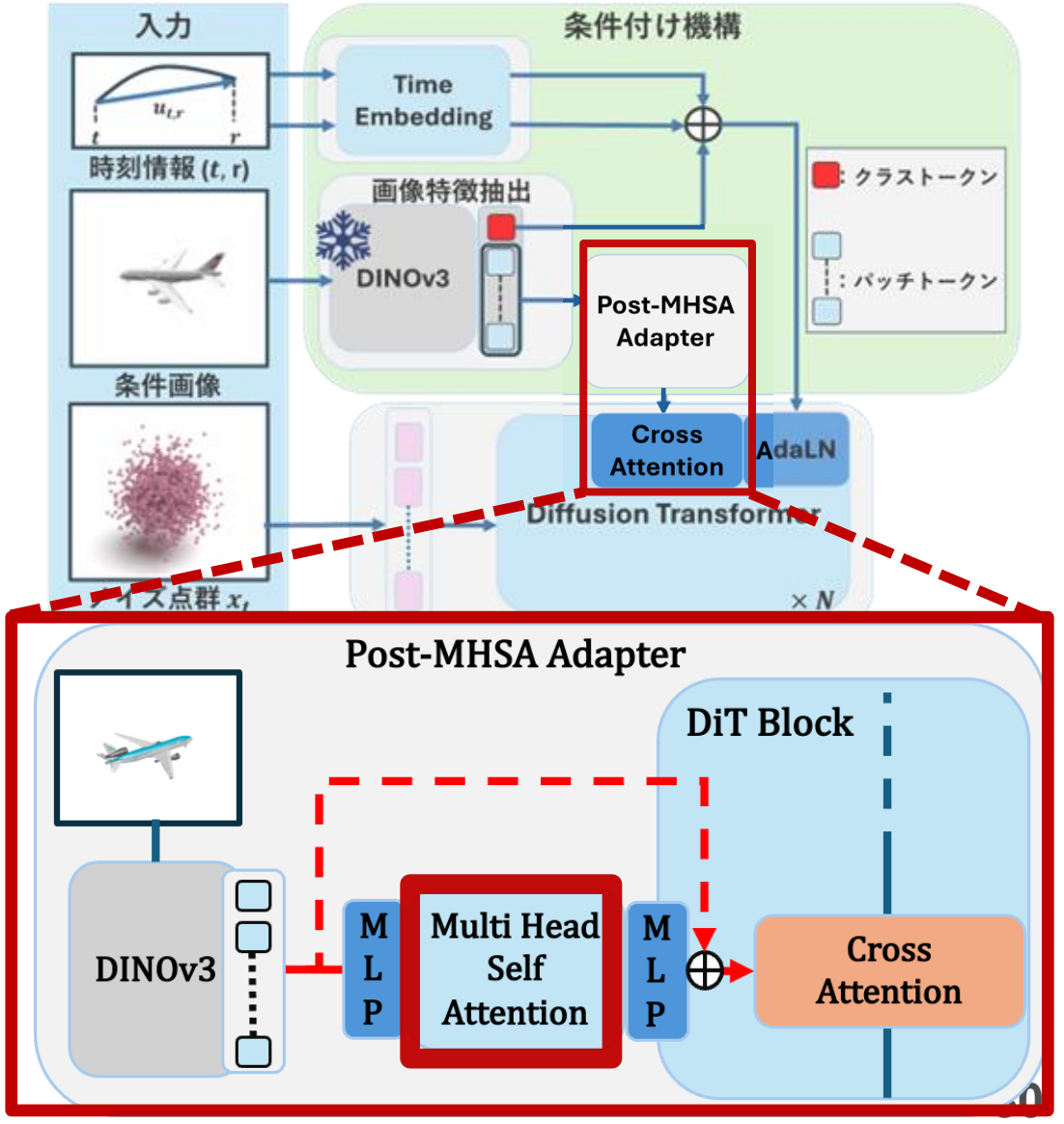
提案手法：Post-MHSA Adapter [新規提案]

- Cross Attentionによる細部情報の入力
  - パッチ特徴は画像細部の特徴を持つが…
    - 点群特徴に最適化されていない可能性
    - 背景などから不要な情報が入る可能性
      - Self Attentionで重要領域を計算
      - DiTの各層のMLPでタスク適応
      - 元の特徴を保つため, 残差接続



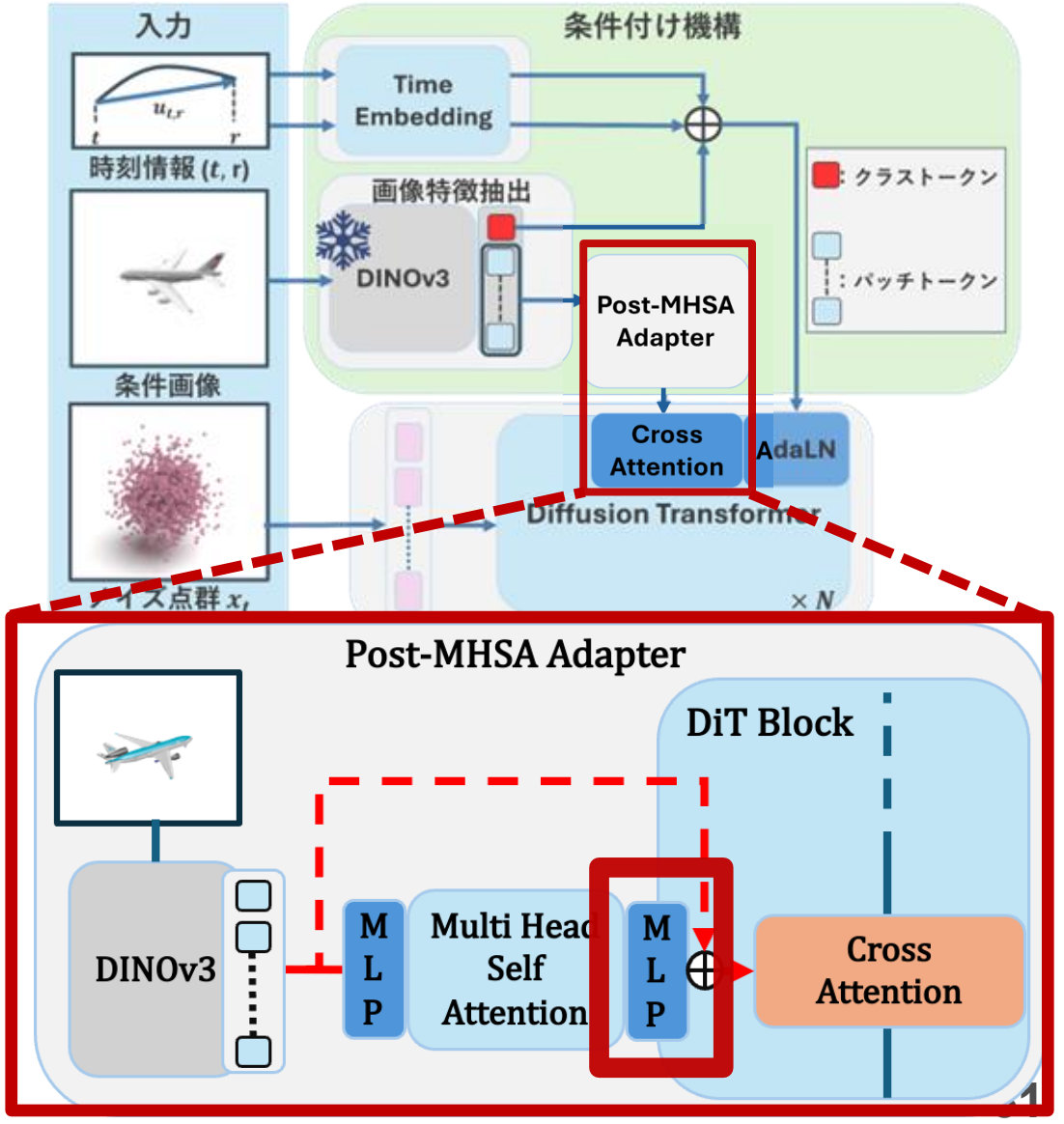
提案手法：Post-MHSA Adapter [新規提案]

- Cross Attentionによる細部情報の入力
  - パッチ特徴は画像細部の特徴を持つが…
    - 点群特徴に最適化されていない可能性
    - 背景などから不要な情報が入る可能性
      - **Self Attentionで重要領域を計算**
      - **DiTの各層のMLPでタスク適応**
      - **元の特徴を保つため, 残差接続**



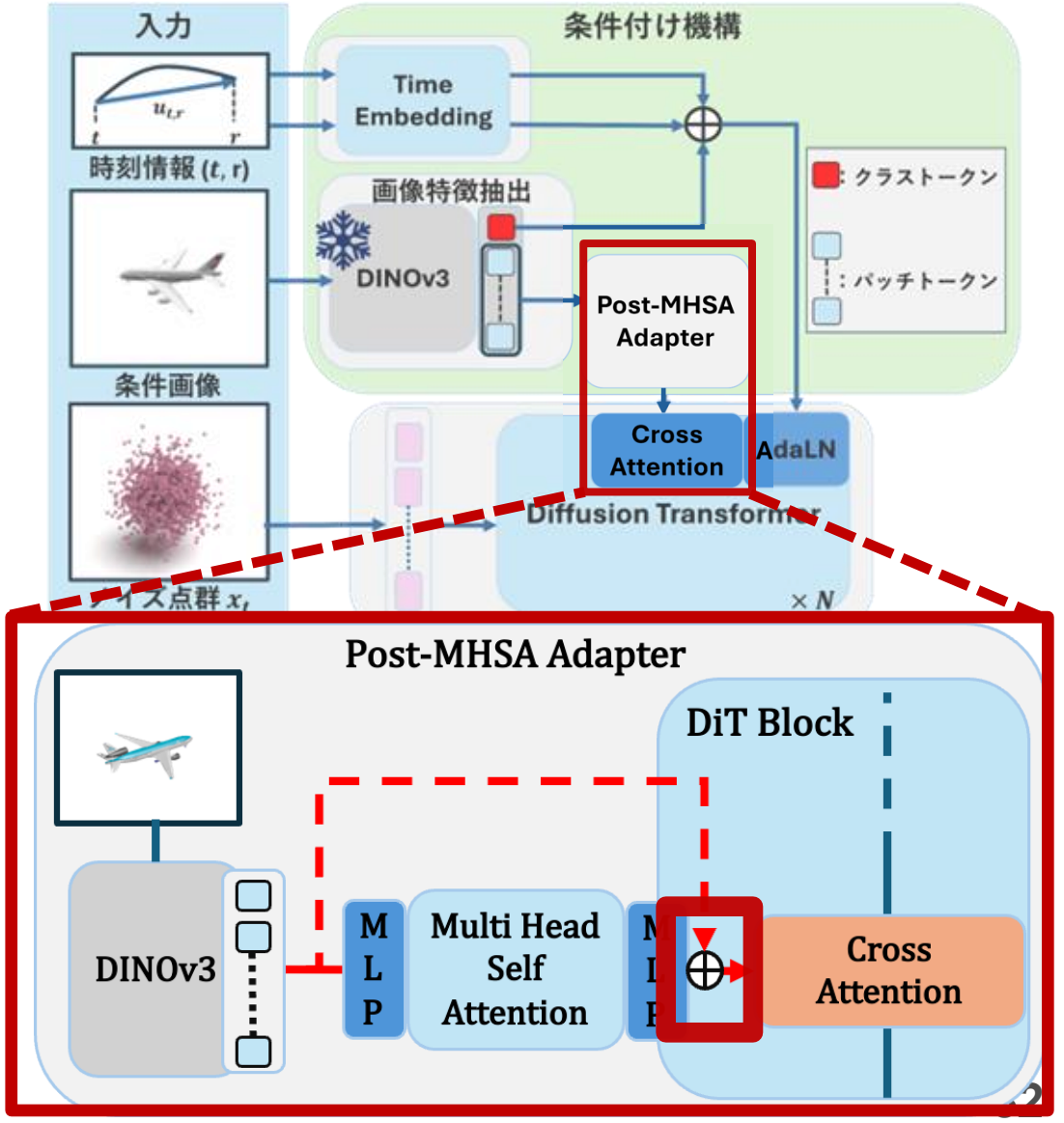
提案手法：Post-MHSA Adapter [新規提案]

- Cross Attentionによる細部情報の入力
  - ➢ パッチ特徴は画像細部の特徴を持つが…
    - ▷ 点群特徴に最適化されていない可能性
    - ▷ 背景などから不要な情報が入る可能性
      - Self Attentionで重要領域を計算
      - DiTの各層のMLPでタスク適応
      - 元の特徴を保つため, 残差接続



提案手法：Post-MHSA Adapter [新規提案]

- Cross Attentionによる細部情報の入力
  - ➢ パッチ特徴は画像細部の特徴を持つが…
    - ▷ 点群特徴に最適化されていない可能性
    - ▷ 背景などから不要な情報が入る可能性
      - Self Attentionで重要領域を計算
      - DiTの各層のMLPでタスク適応
      - 元の特徴を保つため, 残差接続



提案手法： Classifier Free Guidance (CFG) Mean Flow

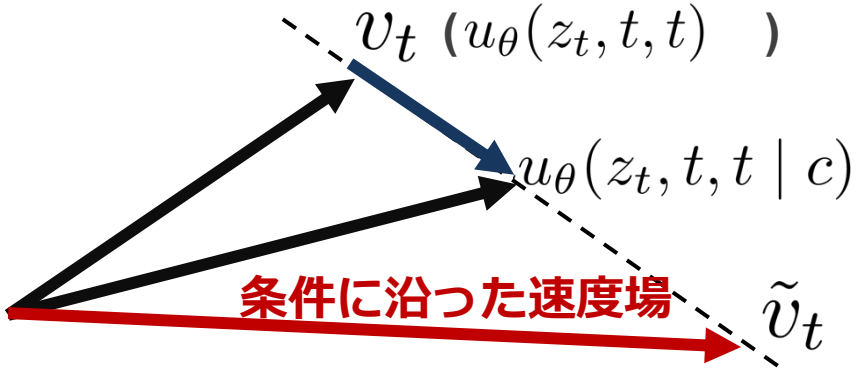
❑ 画像から目的の点群を得るため, 条件付けを行う

AdaLN, Cross Attentionでの画像条件

➢ 条件付け瞬間速度場を定義する

$$\tilde{v}_t := \omega v_t + \kappa u_\theta(z_t, t, t \mid c) + (1 - \omega - \kappa) u_\theta(z_t, t, t)$$

▷ ωは教師速度場$e - x$の強度, κは条件付けの強度



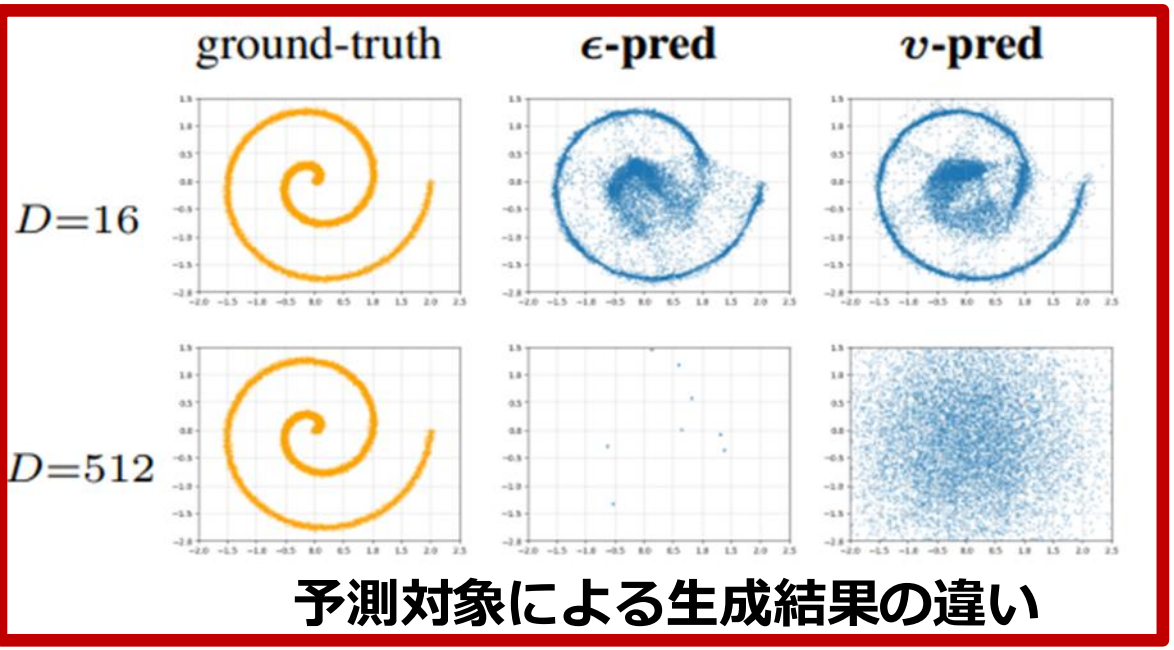
➢ 基本式 $u(z_t, r, t) = \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, d\tau$ を用いて, MeanFlowの損失が求まる.

$$\mathcal{L}_{\text{MF-CFG}}(\theta) = \mathbb{E}\left[\left\| u_\theta(z_t, r, t \mid c) - \text{sg}(u_{\text{tgt}}) \right\|_2^2 \right]$$

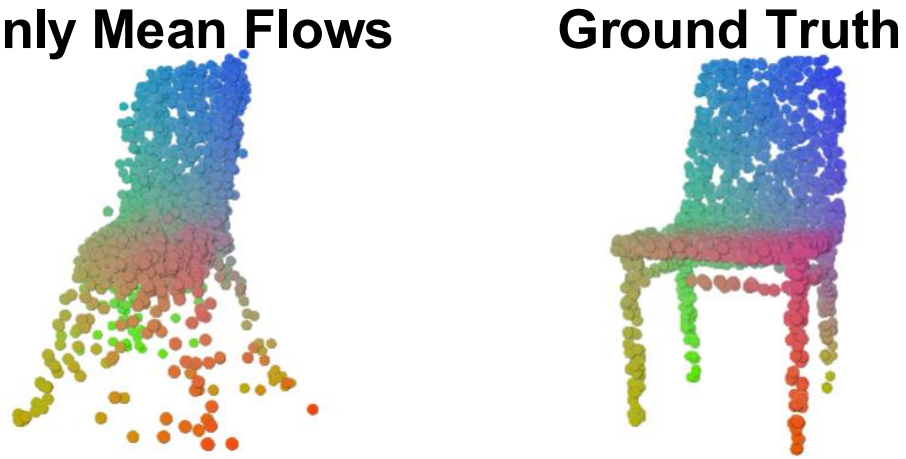
目的の平均速度場は勾配を停止

提案手法： Geometry Nose Anchor (GNA) [新規提案]

- **速度場予測による点群生成では, いくつか問題がある**
  - **高次元データ** (点数=トークン数) なので, 速度場による予測が適さない可能性
  - MSEのみの損失では, 集合全体としての整列を考慮できない可能性



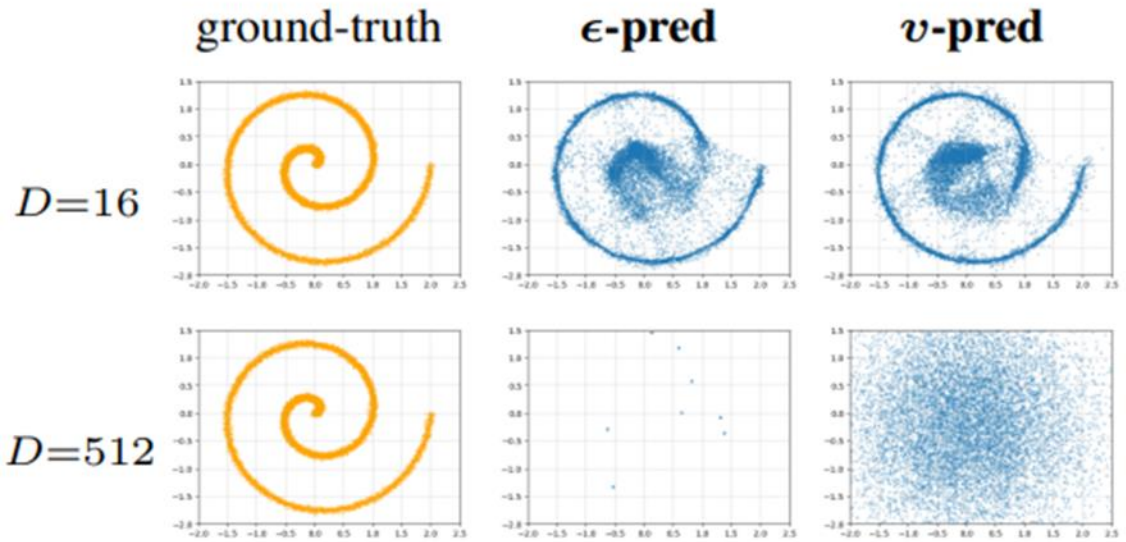
予測対象による生成結果の違い



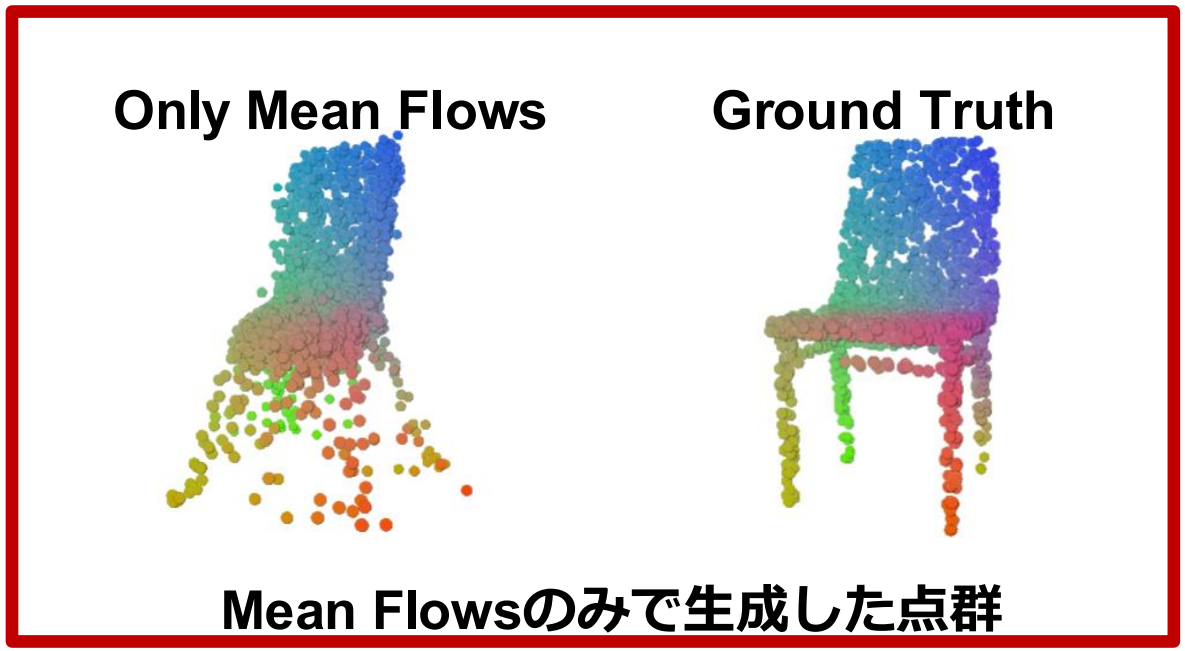
Mean Flowsのみで生成した点群

提案手法： Geometry Nose Anchor (GNA) [新規提案]

- **速度場予測による点群生成では, いくつか問題がある**
  - **高次元データ** (点数=トークン数) なので, 速度場による予測が適さない可能性
  - MSEのみの損失では, 集合全体としての整列を考慮できない可能性



予測対象による生成結果の違い

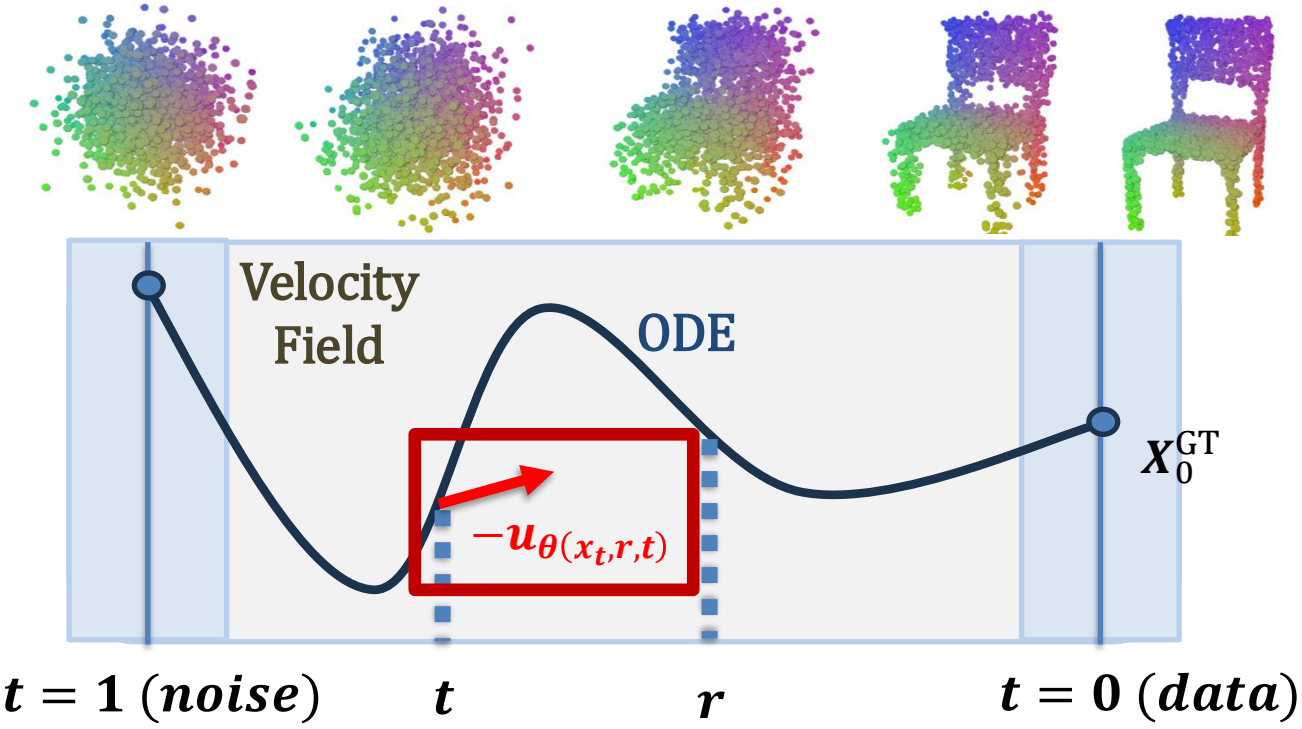


Mean Flowsのみで生成した点群

提案手法： Geometry Nose Anchor (GNA) [新規提案]

❑ 時刻tでの予測デノイズ点群x_θを x_r^{gt}に近づける

➢ 点群間の1対1の距離を確率的に計算するAPML [20] を用いる



➢ 最終的な損失：

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{MF-CFG}} + s\,\lambda(t)\,L_{\text{APML}}\left(x_\theta, x_0^{\text{GT}}\right)$$

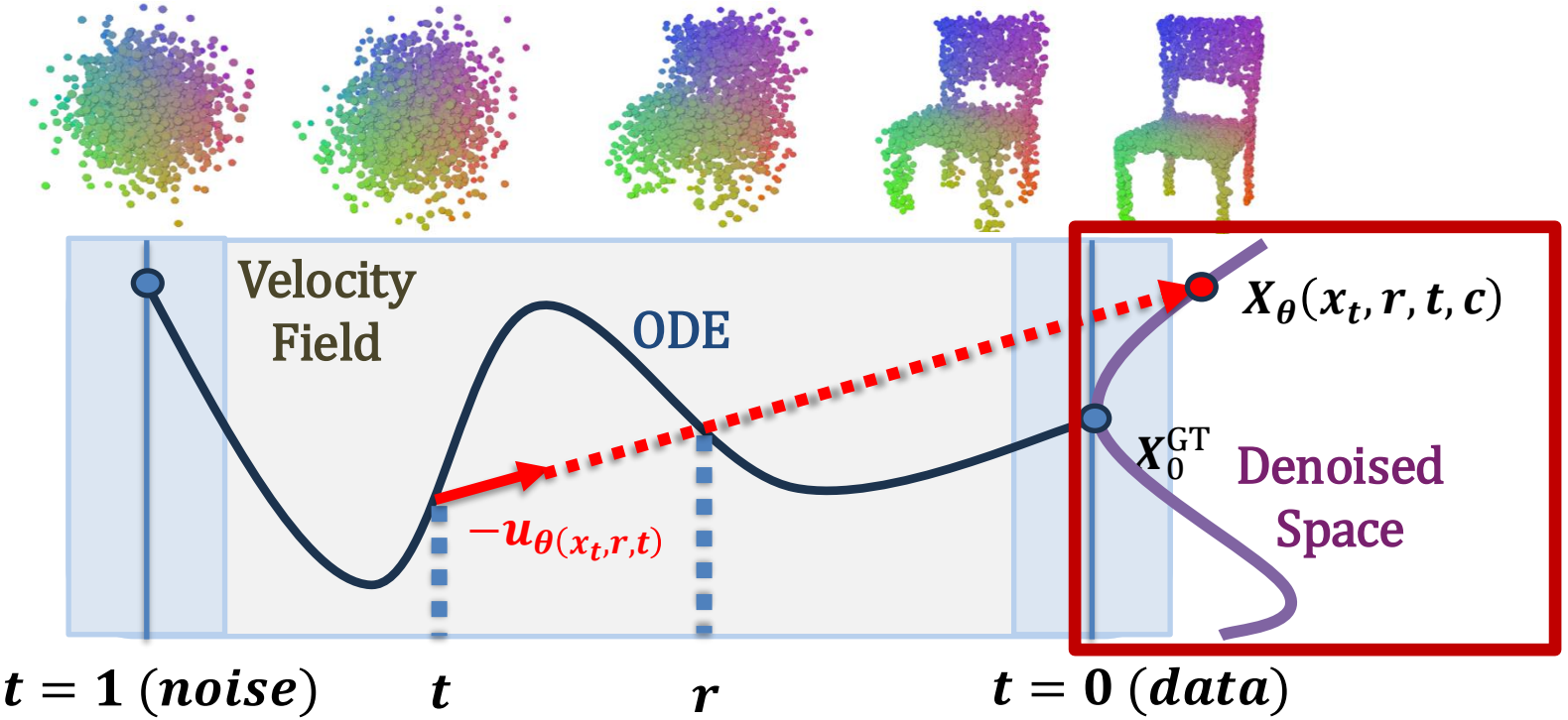
▷ sは主損失とのスケール定数, $\lambda(t)$は時刻に応じて強度を調整するハイパーパラメータ

[20] Sasan Sharifipour et al. "APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction." NeurIPS, 2025.

提案手法： Geometry Nose Anchor (GNA) [新規提案]

❑ 時刻tでの予測デノイズ点群x_θを x_r^{gt}に近づける

➢ 点群間の1対1の距離を確率的に計算するAPML [20] を用いる



➢ 最終的な損失：

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{MF-CFG}} + s\,\lambda(t)\,L_{\text{APML}}\left(x_\theta, x_0^{\text{GT}}\right)$$

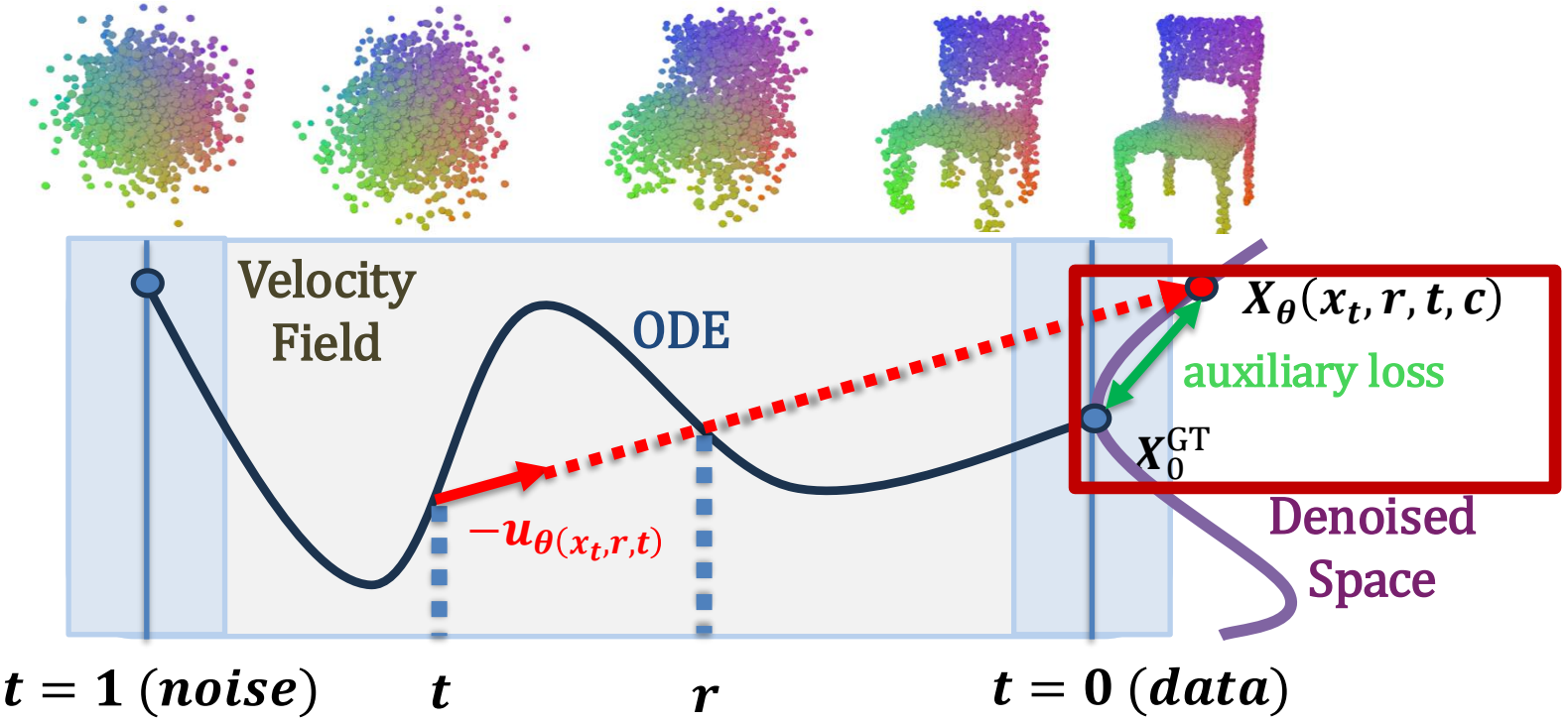
▷ sは主損失とのスケール定数, $\lambda(t)$は時刻に応じて強度を調整するハイパーパラメータ

[20] Sasan Sharifipour et al. "APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction." NeurIPS, 2025.

提案手法： Geometry Nose Anchor (GNA) [新規提案]

❑ 時刻tでの予測デノイズ点群x_θを x_r^{gt}に近づける

➢ 点群間の1対1の距離を確率的に計算するAPML [20] を用いる



➢ 最終的な損失：

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{MF-CFG}} + s\,\lambda(t)\,L_{\text{APML}}\left(x_\theta, x_0^{\text{GT}}\right)$$

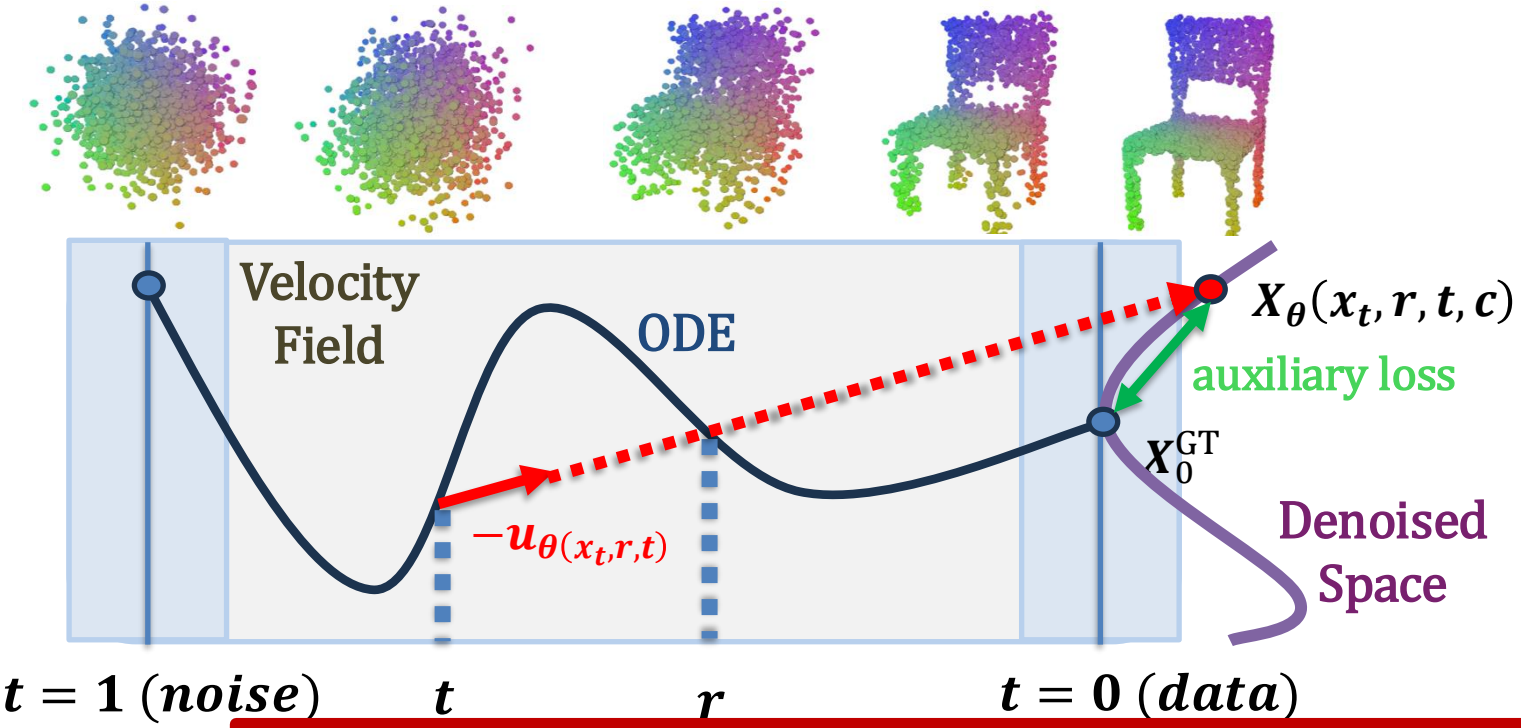
▷ sは主損失とのスケール定数, $\lambda(t)$は時刻に応じて強度を調整するハイパーパラメータ

[20] Sasan Sharifipour et al. "APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction." NeurIPS, 2025.

提案手法： Geometry Nose Anchor (GNA) [新規提案]

❑ 時刻tでの予測デノイズ点群x_θを x_r^{gt}に近づける

➢ 点群間の1対1の距離を確率的に計算するAPML [20] を用いる



➢ 最終的な損失：

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{MF-CFG}} + s\,\lambda(t)\,L_{\text{APML}}\left(x_\theta, x_0^{\text{GT}}\right)$$

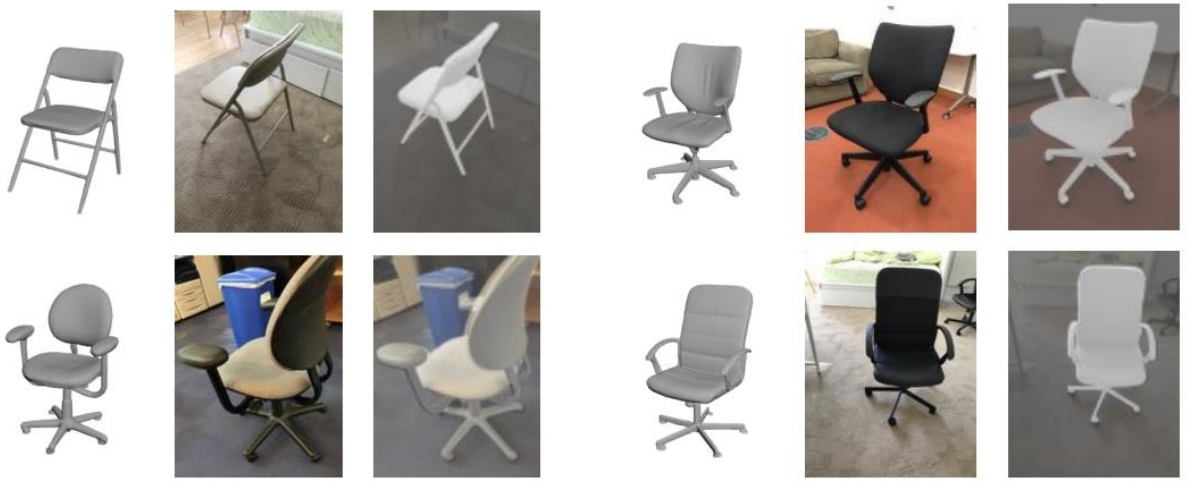
▷sは主損失とのスケール定数, $\lambda(t)$は時刻に応じて強度を調整するハイパーパラメータ

[20] Sasan Sharifipour et al. "APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction." NeurIPS, 2025.

実験：実験設定

- **データセット**
  - **合成データセット：** ShapeNet [21]
  - **実世界データセット**：Pix3D [*]
- **評価指標**
  - Chamfer Distance (CD)
  - Earth Mover’s Distance (EMD)
    - **どちらも, 下がるほど良い**

ShapeNetの一部



Pix3Dの一部

[21] Angel X. Chang et al. "ShapeNet: An Information-Rich 3D Model Repository." arXiv:1512.03012, 2015.
[*] Xingyuan Sun et al. "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling." In CVPR, 2018.

実験：定量評価 (ShapeNet)

❑CD, EMDでの比較

表1：ShapeNetにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Eearth Mover's Distance (×100) ↓			
	Car	Chair	Air	Avg.	Car	Chair	Air	Avg.
Self-Sup. (2020)	5.48	10.91	7.11	7.11	4.95	14.93	11.07	10.31
DIFFER (2019)	6.35	9.78	5.67	7.27	6.03	16.21	9.9	10.71
ULSP (2018)	5.4	9.72	5.91	7.01	4.78	10.18	7.66	7.54
RGB2point (2025)	**4.22**	5.43	**2.7**	<u>4.33</u>	5.63	9.53	6.86	7.83
PC^2 (256-NFE) (2023)	<u>4.95</u>	5.6	3.37	4.84	4.69	5.36	4.27	4.84
BDM (276-NFE) (2024)	5.12	<u>5.02</u>	3.12	4.64	<u>4.25</u>	<u>4.53</u>	<u>3.92</u>	<u>4.28</u>
Ours (1-NFE)	**4.22**	**4.29**	<u>2.76</u>	**3.92**	**3.82**	**4.29**	**3.04**	**3.82**

➢ 既存手法をCDで約9.5%, EMDで約11%上回った.

実験：定量評価 (ShapeNet)

❑CD, EMDでの比較

表1：ShapeNetにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Eearth Mover's Distance (×100) ↓			
	Car	Chair	Air	Avg.	Car	Chair	Air	Avg.
Self-Sup. (2020)	5.48	10.91	7.11	7.11	4.95	14.93	11.07	10.31
DIFFER (2019)	6.35	9.78	5.67	7.27	6.03	16.21	9.9	10.71
ULSP (2018)	5.4	9.72	5.91	7.01	4.78	10.18	7.66	7.54
RGB2point (2025)	**4.22**	5.43	**2.7**	<u>4.33</u>	5.63	9.53	6.86	7.83
PC^2 (256-NFE) (2023)	<u>4.95</u>	5.6	3.37	4.84	4.69	5.36	4.27	4.84
BDM (276-NFE) (2024)	5.12	<u>5.02</u>	3.12	4.64	<u>4.25</u>	<u>4.53</u>	<u>3.92</u>	<u>4.28</u>
Ours (1-NFE)	**4.22**	**4.29**	<u>2.76</u>	**3.92**	**3.82**	**4.29**	**3.04**	**3.82**

- 既存手法をCDで約9.5%, EMDで約11%上回った.

実験：定量評価 (ShapeNet)

❑CD, EMDでの比較

表1：ShapeNetにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Eearth Mover's Distance (×100) ↓			
	Car	Chair	Air	Avg.	Car	Chair	Air	Avg.
Self-Sup. (2020)	5.48	10.91	7.11	7.11	4.95	14.93	11.07	10.31
DIFFER (2019)	6.35	9.78	5.67	7.27	6.03	16.21	9.9	10.71
ULSP (2018)	5.4	9.72	5.91	7.01	4.78	10.18	7.66	7.54
RGB2point (2025)	**4.22**	5.43	**2.7**	<u>4.33</u>	5.63	9.53	6.86	7.83
PC^2 (256-NFE) (2023)	<u>4.95</u>	5.6	3.37	4.84	4.69	5.36	4.27	4.84
BDM (276-NFE) (2024)	5.12	<u>5.02</u>	3.12	4.64	<u>4.25</u>	<u>4.53</u>	<u>3.92</u>	<u>4.28</u>
Ours (1-NFE)	**4.22**	**4.29**	<u>2.76</u>	**3.92**	**3.82**	**4.29**	**3.04**	**3.82**

- 既存手法をCDで約9.5%, EMDで約11%上回った.

実験：定量評価 (ShapeNet)

❑CD, EMDでの比較

表1：ShapeNetにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Eearth Mover's Distance (×100) ↓			
	Car	Chair	Air	Avg.	Car	Chair	Air	Avg.
Self-Sup. (2020)	5.48	10.91	7.11	7.11	4.95	14.93	11.07	10.31
DIFFER (2019)	6.35	9.78	5.67	7.27	6.03	16.21	9.9	10.71
ULSP (2018)	5.4	9.72	5.91	7.01	4.78	10.18	7.66	7.54
RGB2point (2025)	**4.22**	5.43	**2.7**	<u>4.33</u>	5.63	9.53	6.86	7.83
PC^2 (256-NFE) (2023)	<u>4.95</u>	5.6	3.37	4.84	4.69	5.36	4.27	4.84
BDM (276-NFE) (2024)	5.12	<u>5.02</u>	3.12	4.64	<u>4.25</u>	<u>4.53</u>	<u>3.92</u>	<u>4.28</u>
Ours (1-NFE)	**4.22**	**4.29**	<u>2.76</u>	**3.92**	**3.82**	**4.29**	**3.04**	**3.82**

➢ 既存手法をCDで約9.5%, EMDで約11%上回った.

-9.4%

-11%

実験：定量評価 (Pix3D)

❑ CD, EMDでの比較

表1：Pix3Dにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Earth Mover's Distance (×100) ↓			
	Chair	Sofa	Table	Avg.	Chair	Sofa	Table	Avg.
PC^2 (256-NFE) (2023)	7.07	5.87	11.39	7.82	7.13	6.14	10.72	7.75
BDM (276-NFE) (2024)	<u>6.62</u>	5.55	8.35	7.12	<u>7.05</u>	**4.98**	<u>8.79</u>	<u>7.12</u>
RGB2point (2025)	<u>6.62</u>	<u>5.4</u>	<u>9.7</u>	<u>7.06</u>	9.34	7.11	12.69	9.59
Ours (1-NFE)	**6.48**	**5.02**	**8.2**	**6.53**	**6.74**	<u>5.06</u>	**7.94**	**6.61**

➢ 既存手法をCDで約7.5%, EMDで約7.1%上回った.

実験：定量評価 (Pix3D)

❑ CD, EMDでの比較

表1：Pix3Dにおける既存手法との定量評価 (CD/EMD，各×100，**赤字**：最良値，<u>下線</u>：次点)

Method	Chamfer Distance (×100) ↓				Earth Mover's Distance (×100) ↓			
	Chair	Sofa	Table	Avg.	Chair	Sofa	Table	Avg.
PC^2 (256-NFE) (2023)	7.07	5.87	11.39	7.82	7.13	6.14	10.72	7.75
BDM (276-NFE) (2024)	<u>6.62</u>	5.55	8.35	7.12	<u>7.05</u>	4.98	<u>8.79</u>	<u>7.12</u>
RGB2point (2025)	<u>6.62</u>	<u>5.4</u>	<u>9.7</u>	<u>7.06</u>	[illegible]34	7.11	12.69	9.59
Ours (1-NFE)	6.48	5.02	8.2	6.53	[illegible]74	<u>5.06</u>	7.94	6.61

-7%

-7%

➢ 既存手法をCDで約7.5%, EMDで約7.1%上回った.

実験：生成効率

❑ 実行速度＆VRAM

➢ 点数を揃えた拡散モデルベース，フィードフォワードベースとの比較

表2：推論時間とVRAMピーク使用量の比較

Method	Time [ms/sample]	VRAM [GiB]	CD (×100) ↓	EMD (×100) ↓
PC^2 (256-NFE)	22000 ± 200	1.73	4.84	4.84
BDM (276-NFE)	28000 ± 200	1.97	4.64	<u>4.28</u>
RGB2point	28.39 ± 2.18	0.818	<u>4.33</u>	7.83
Ours (1-NFE)	63.45 ± 0.25	1.282	3.92	3.82

▷フィードフォワードよりはやや遅いが，拡散モデルベースの**347倍**高速

実験：生成効率

❑ 実行速度＆VRAM

➢ 点数を揃えた拡散モデルベース，フィードフォワードベースとの比較

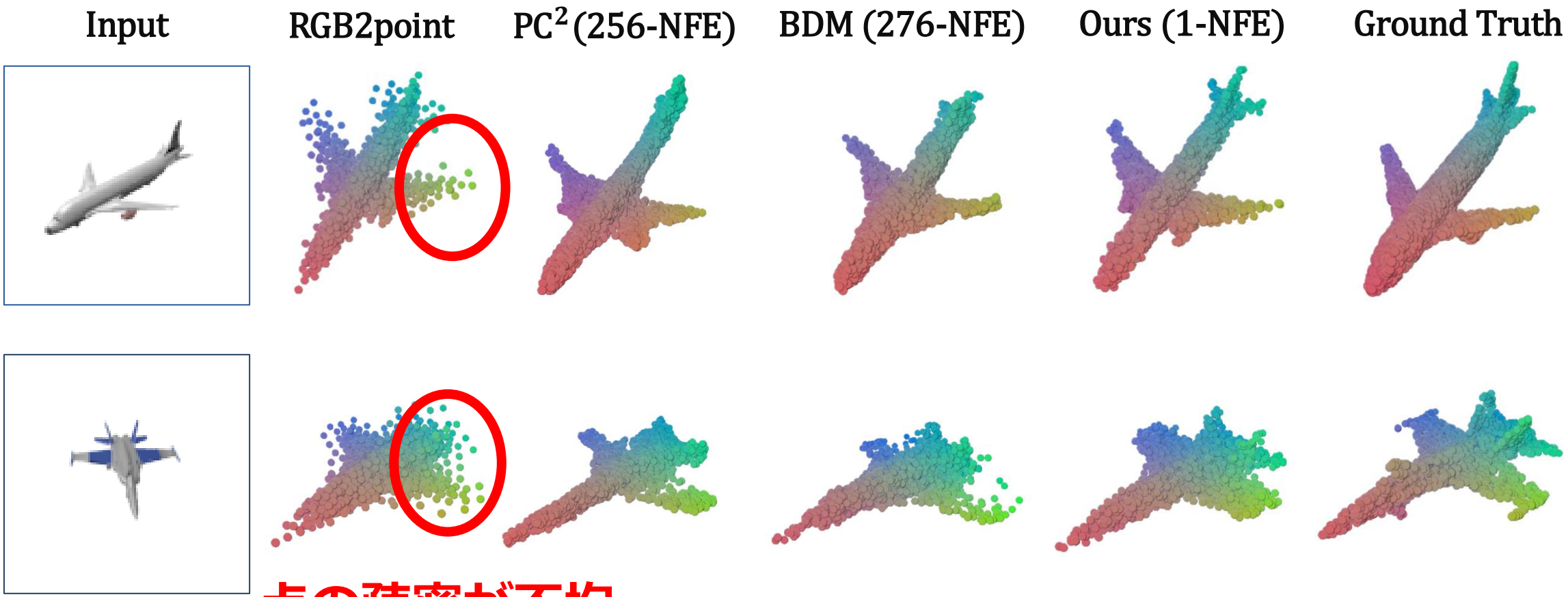
表2：推論時間とVRAMピーク使用量の比較

Method	Time [ms/sample]	VRAM [GiB]	CD (×100) ↓	EMD (×100) ↓
PC^2 (256-NFE)	22000 ± 200	1.73	4.84	4.84
BDM (276-NFE)	28000 ± 200	1.97	4.64	4.28
RGB2point	28.39 ± 2.18	0.818	4.33	7.83
Ours (1-NFE)	63.45 ± 0.25	1.282	3.92	3.82

347倍

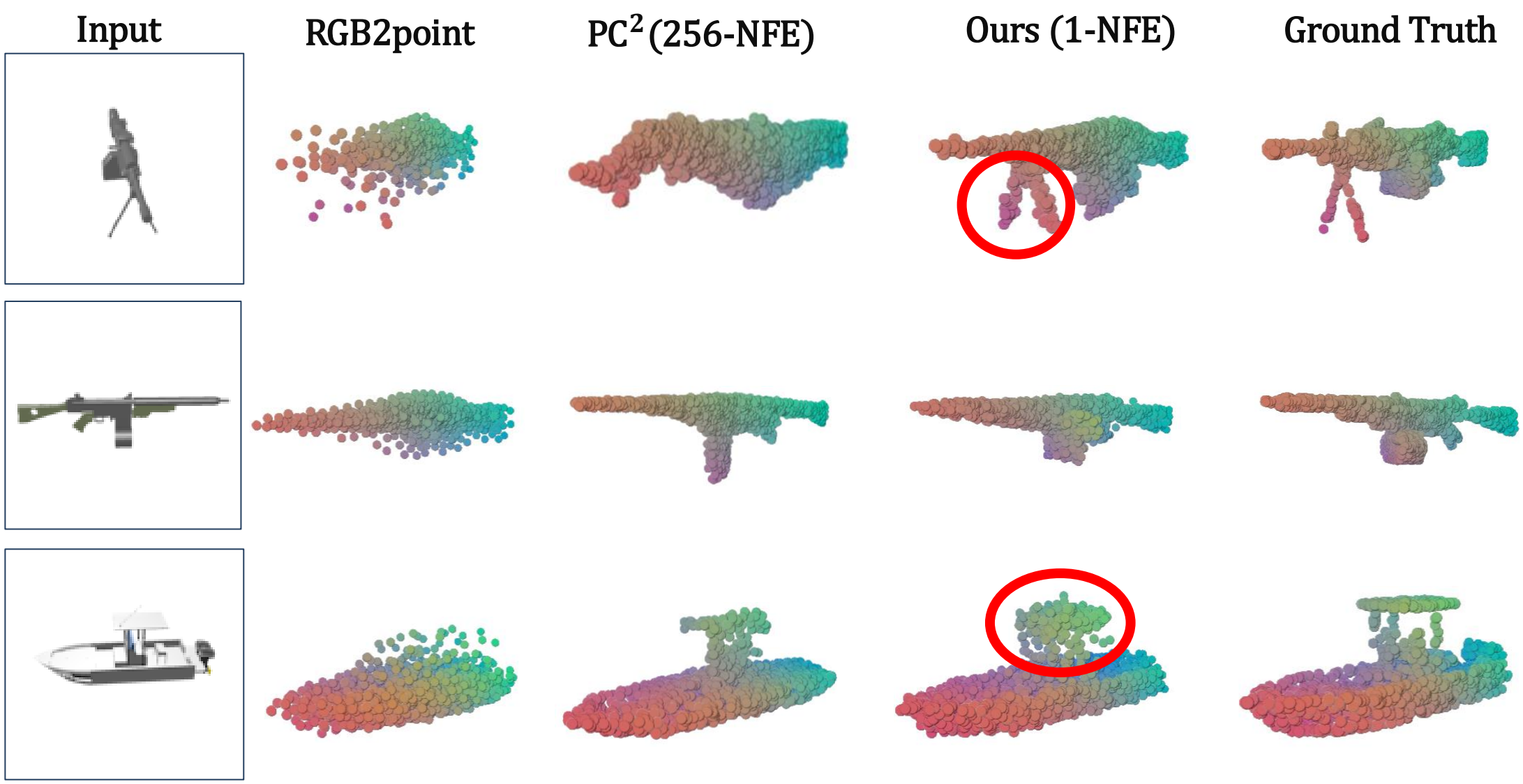
▷フィードフォワードよりはやや遅いが，拡散モデルベースの**347倍**高速

実験：定性評価 (ShapeNet)

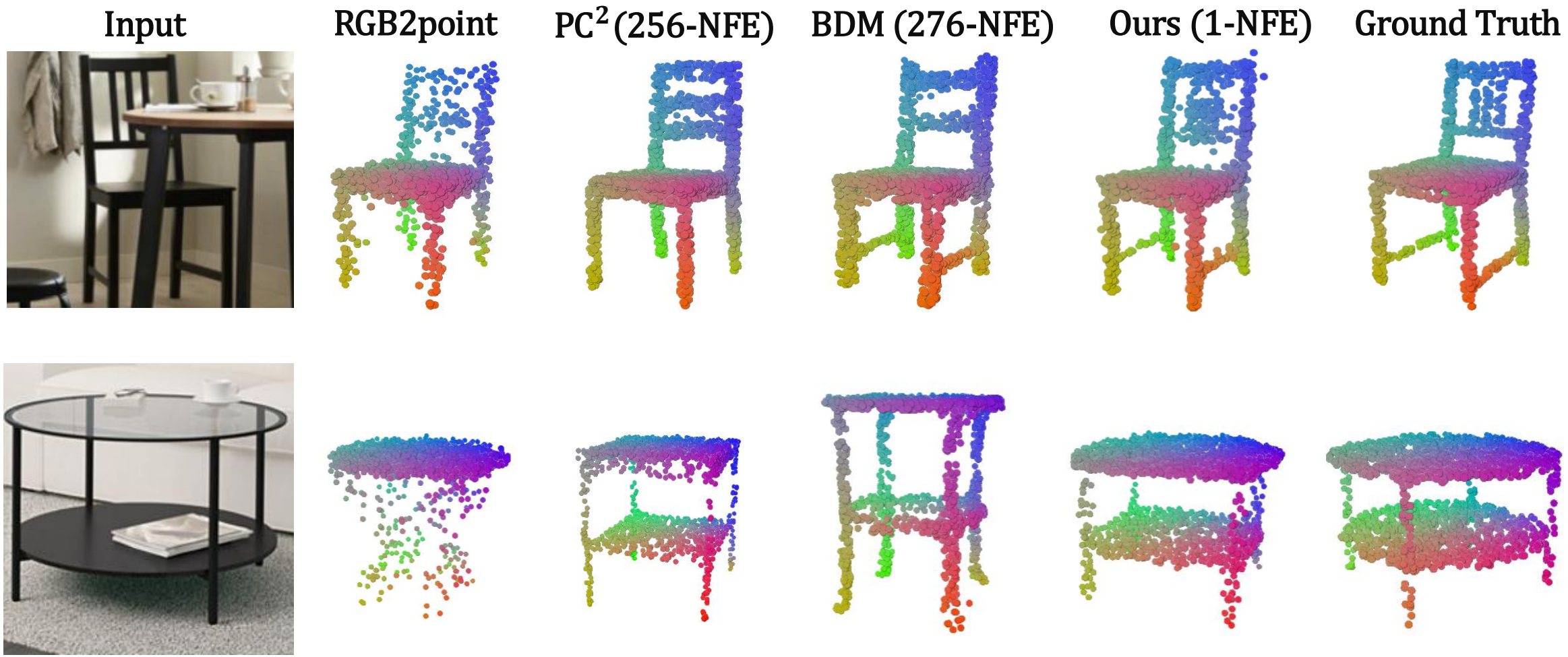


Aircraft の生成結果

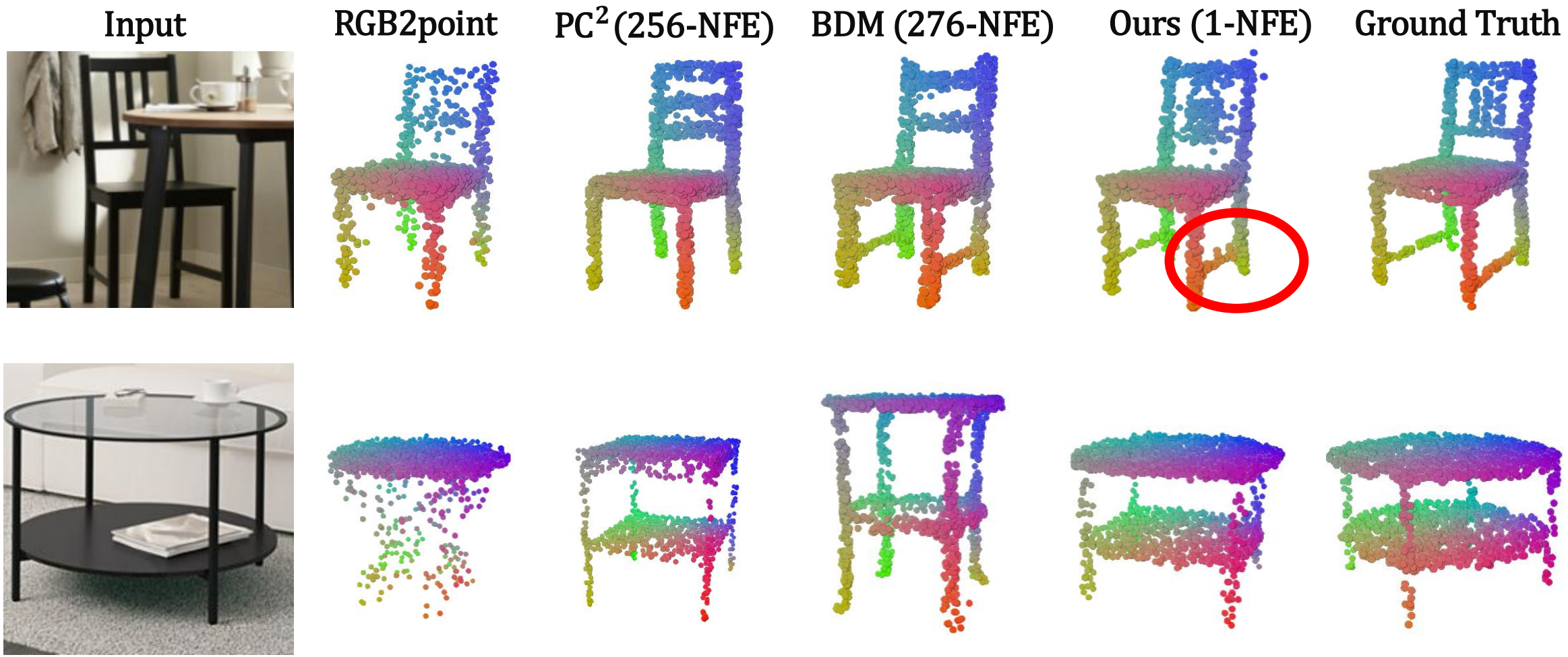
実験：定性評価 (ShepeNet)



実験：定性評価 (Pix3D)



実験：定性評価 (Pix3D)



まとめ

❑ 主なポイント

- **点群生成分野において, 初めてMean Flowを応用した.**
- **Chamfer Distance, Earth Mover's Distanceで既存手法を上回った.**
- **フィードフォワード型モデルの2倍ほどの増加で, 拡散モデルベースでの生成.**

❑ 課題点と展望

- Mean Flowの訓練では, Jacobian-Vector Product (JVP) の高コスト計算が必要
  - **生成は高速だが, 訓練のコストが高い. (本研究ではA6000×8で36時間訓練)**
  - **訓練コストの削減, より複雑なデータセットでの検証**

補足資料①：実装の詳細とアブレーション

❑ 実装詳細

- AdamW / 学習率1e-4 / バッチ128 / 120k step
- DiT：ブロックは12層, 隠れ次元512, 点数は1024
- GNAの重み$\lambda(t,r)$：データ側で大きく (r=0, 0.40), (r=1, 0.20)
- スクラッチ学習 (DINO除き)
- サンプリングはEular法：$x_0 = x_t - (t-0)\,u_\theta(x_t, t, 0, c)$
- ガイダンススケールは$\omega = 1.0, \kappa = 0.5$

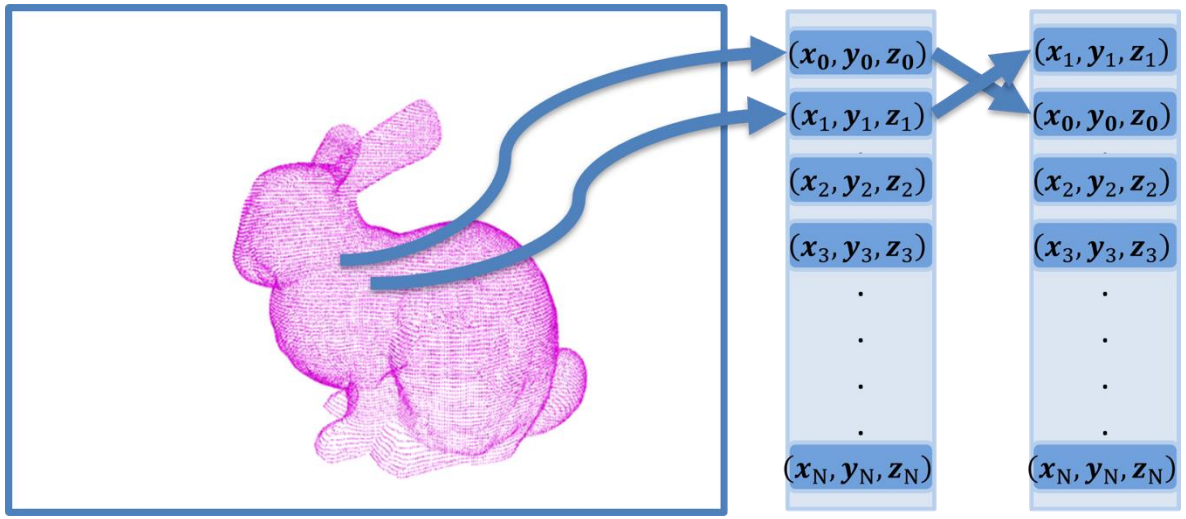
❑ 各提案手法の有効性：w/oは提案部を除去

Method	Chamfer Distance (×100) ↓			Earth Mover's Distance (×100) ↓		
	Car	Chair	Avg.	Car	Chair	Avg.
Ours (1-NFE)	4.22	4.29	3.92	3.82	4.29	3.82
Ours (1-NFE, w/o Adapter)	4.58	4.29	4.12	4.02	4.62	4.13
Ours (1-NFE, w/o GNA)	5.21	6.16	5.23	4.89	6.08	5.16

補足資料②：点群トークンの埋め込み

❑ 点群トークンの埋め込み

- ➢ 点群は並びに意味が無い
- ➢ 並び順に依存しない埋め込みが必要
  - ▷ **トークンの順番に基づく埋め込みを省く**



点群の順列不変性

補足資料③：補助損失のスケール

❑ **最終損失** $\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{MF-CFG}} + s\,\lambda(t)\,L_{\text{APML}}\left(x_\theta, x_0^{\text{GT}}\right)$

➢ $s = \dfrac{\mathbb{E}[\mathcal{L}_{\text{MF-CFG}}]_{\text{detach}}}{\mathbb{E}[\mathcal{L}_{\text{DSA}}]_{\text{detach}} + \delta}$

▷ バッチ平均でスケール差を吸収させる

➢ $\lambda(t) = \mathbf{1}_{t \neq r}\ \dfrac{\lambda_{\text{base}}}{\max(t, \tau)}$

▷ データ側の時刻で線形に大きくなるように設計

• sでスケールを合わせたため，λは主損失の何割を占めるかを設定

補足資料④：アブレーション

❑ 補助損失の距離損失の変更

Method	CD×100 ↓				EMD×100 ↓			
	Chair	Sofa	Table	Mean	Chair	Sofa	Table	Mean
APML	6.48	**5.02**	**8.20**	**6.53**	**6.74**	**5.06**	**7.94**	**6.61**
CD	**6.36**	5.12	8.40	6.54	7.48	5.26	8.53	7.17
MSE	11.18	8.63	12.84	10.93	10.1	9.52	10.21	10.93
w/o GNA	8.82	6.58	10.12	8.58	8.88	7.02	9.76	8.62

➢ **集合距離**を用いることが重要であることがわかる