一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

平均速度場モデルを用いた単一画像からの点群再構成

馬場　雄大†　　柳井　啓司†

† 電気通信大学　情報理工学域　I 類　〒182–8585 東京都調布市調布ヶ丘一丁目 5 番地 1
E-mail: †baba-y@mm.inf.uec.ac.jp, ††yanai@cs.uec.ac.jp

あらまし　単一画像からの点群再構成は不可視領域の補完を要する不良設定であり，高品質化には拡散モデルが有効だが推論コストが課題となる．本研究では Mean Flow に基づき，点群空間で区間平均速度場を直接推定する条件付き Diffusion Transformer の枠組みを提案する．時刻 t と区間長 dt を明示的に条件化し，DINOv3 の大域特徴と, 新規のアダプターを通過した細部特徴を注入する．さらに，1-step のまま条件整合を強める MF-CFG と，復元点群 x_0 を集合距離で拘束する Geometry Noise Anchor を導入し，ShapeNet において最新のフィードフォワード型モデルを CD, EMD, F-Score で上回り, 従来の拡散モデルベースよりも高速に動作することを確認した．

キーワード　3 次元再構成, 点群, フローマッチング

1. ま え が き

単一画像からの点群再構成は，画像に写る可視領域から不可視領域を含む三次元形状を推定する不良設定問題であり，観測の曖昧さに起因する多様な解を扱う必要がある．この課題は AR/VR における空間理解・提示品質の向上や，ロボティクスにおける環境認識（把持，ナビゲーション等）の基盤として重要であり，実運用では低遅延な推論が求められる．

点群は前処理が軽く解像度制御が容易なため，再構成の中間表現としても有用である．例えば, 各点に属性（スケール，回転，不透明度，色など）を付与することで，3D Gaussian Splatting [1] の基礎要素へ拡張でき，レンダリングや新規視点合成へ接続しやすい．

近年は拡散モデル [2] や Flow Matching [3] が不確実性を伴う生成の枠組みとして注目される一方，推論時に多数回の反復更新（Network Function Evaluation, NFE）を要し，遅延・消費電力が課題となる．Mean Flow [4] は区間平均の更新量（平均速度場）を直接推定して少 NFE 生成を狙うが，主な実証は事前学習済み VAE で得た潜在表現上で行われている点に注意が必要である．潜在空間での生成は VAE の再構成性能に依存し，追加学習も必要となる．また点群を潜在化する場合，点群エンコーダ/デコーダは一般に近傍探索を伴う点演算 [5], [6] や疎な三次元畳み込み [7] を組み合わせた複雑な構成になりやすく，実装・最適化の負担が増す可能性がある．実際, このような動的な処理は ONNX, CoreML などのエッジデバイス向けの変換は困難である．

そこで本研究では，VAE による潜在化を用いず点群空間で Mean Flow に基づく生成過程を設計し，少 NFE な単一画像条件付き点群再構成を目指す．点群空間の少ステップ生成では区間ジャンプが大きく，多様体からの逸脱や外れ点が生じやすい．本研究ではこれを抑えるため，速度 v ではなく復元点群 x_0 に対する集合距離に基づく補助損失を導入し，Just image Transformer (JiT) [8] が指摘する多様体仮説に基づいて安定化を図る．以上より，本研究の狙いは (i) Mean Flow に基づく少 NFE 生成の適用，(ii) 点群空間での直接生成，(iii) x_0 拘束による安定化，により低遅延かつ実装容易な再構成系を構築することである．

2. 関 連 研 究

2.1 単一画像からの 3 次元再構成

単一画像から三次元形状を復元する研究では，ボクセル，メッシュ，点群など多様な表現が検討されてきた．例えば 3D-R2N2 [9] は 2D 特徴を 3D Convolutional LSTM で統合し，ボクセル占有を逐次的に更新する．また，姿勢推定や微分可能レンダリングにより 2D 整合性から学習信号を得る手法も提案されている [10], [11].

点群出力では，再投影一致などで観測整合性を強める工夫が報告されている [12]．RGB2Point [13] は ImageNet 事前学習 ViT を用いた Transformer で点群を直接生成し，姿勢不要かつ簡潔な構成で高速・低 VRAM な推論を実現する．一方で，この種の直接生成は決定的推定に寄りやすく，不確実性の表現という点では拡散モデル等の確率的枠組みが依然重要である．

2.2 拡散モデルによる点群生成と潜在表現

近年は，拡散モデルに代表される確率的生成が，点群生成・再構成にも適用されている．Point-E [14] は，画像条件として凍結した CLIP のグリッド特徴を用い，Transformer により点群を逐次生成する点群拡散モデルである．一方，LION [15] は点群を直接拡散するのではなく，階層 VAE で潜在表現へ写像し，潜在空間上で拡散を行ってから復元する枠組みを採用する．潜在拡散は分布の学習を平滑な潜在空間へ移し替えられる利点がある反面，基本的に両手法ともに反復サンプリングを前提とするため，推論時の NFE が計算コストのボトルネックになり得る．

2.3 Flow Matching と Mean Flow

拡散モデルの高速化として，Flow Matching（FM）[3] は，連続時間の生成を速度場の学習として定式化し，少ない NFE で

の生成を狙う枠組みである．FM の学習は，ある確率パス $\{p_t\}$ を生成する真の速度場 $u_t(x)$ に対して，近似速度場 v_θ を二乗誤差で回帰する．

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

しかし一般に，p_t からのサンプリングや $u_t(x)$ の評価は難しい．そこで Conditional Flow Matching（CFM）では，データ終点 $x_1 \sim q$ に条件付けた条件付き確率パス $p_t(x \mid x_1)$ と条件付き速度場 $u_t(x \mid x_1)$ を設計して，以下を最小化する．

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t,\, x_1\sim q,\, x\sim p_t(\cdot\mid x_1)}\left[\|v_\theta(x,t) - u_t(x \mid x_1)\|_2^2\right] \tag{2}$$

本研究で用いる代表例として，独立結合（independent coupling）に基づく線形補間を考える．

$$x_0 \sim \mathcal{N}(0, I), \quad x_1 \sim q, \quad x_t = (1-t)x_0 + t x_1 \tag{3}$$

式 (3) を用いることで，教師信号は $(x_1 - x_0)$ に簡約されるため，実際に学習可能な損失は以下のようにして，学習する．

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\left[\|v_\theta(x_t,t) - (x_1 - x_0)\|_2^2\right] \tag{4}$$

ただし，条件付き確率パスを直線的に設計しても，周辺速度場は期待値で定義されるため解軌道が曲がりやすく，粗い離散化（低 NFE）では数値積分誤差が累積しやすいことが指摘されている．Mean Flow（MF）[4] はこの課題に対し，区間平均速度場を直接近似することで図 1 のように大きな時間間隔の更新を 1 回または少数回の NFE で近似する枠組みを導入する．

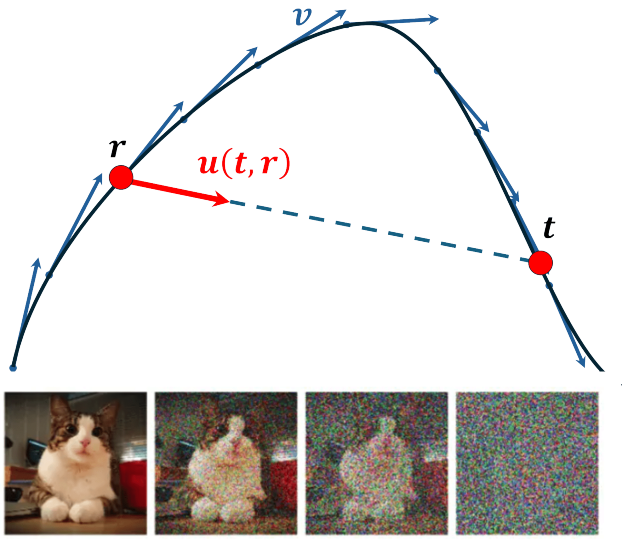

図 1　Mean Flow のイメージ

FM の瞬間速度場を $v(\cdot)$ とし，$0 \le r < t \le 1$ で区間 $[r,t]$ にわたる平均速度場を以下のように定義する．

$$u(x_t, r, t) = \frac{1}{t-r}\int_r^t v(x_\tau, \tau)\, d\tau \tag{5}$$

定義式 (5) より，区間ジャンプは以下で与えられる．

$$x_r = x_t - (t-r)\, u(x_t, r, t) \tag{6}$$

従って u を直接近似できれば，大きな時間間隔を少ない NFE で更新できる．本研究はこの MF の利点を単一画像条件付き点群再構成へ展開し，低遅延な生成を目指す．

2.4　予測空間と多様体仮説

拡散・フロー系モデルでは，ネットワークが予測するノイズ ϵ，速度場 v，ノイズ除去データ x の選択が学習安定性や高次元データでの挙動に影響することが議論されている．JiT は，実データが低次元多様体上にあるという仮説の下で，高次元入力に対する予測空間の選択の重要性を指摘している．本研究では，点群を VAE で潜在化せず point space で直接生成過程を設計する方針の下，補助損失を速度 v ではなく復元点群 x_0 に対して定義し，多様体からの逸脱を抑える整合性拘束として導入する．これにより，低 NFE 生成で問題となりやすい誤差の増幅を抑えつつ，点群空間での直接生成を安定化させることを狙う．

3.　提 案 手 法

図 2 に提案法の全体構成を示す．本研究では単一画像条件付き点群生成を少 NFE で実現するため，点群をトークン列として処理する Diffusion Transformer（DiT）[16] を採用する．入力は時刻 t の点群状態 x_t であり，画像から抽出した条件特徴と，時刻 t および区間長 $dt = t - r$ を統合して，平均速度場 $u_\theta(x_t, r, t \mid c)$ を推定する．ベースは Lan ら [17] の疎点群向け DiT 構成に従いつつ，Mean Flow の区間平均速度場推定に適するよう条件注入を再設計した．

3.1　点群トークン表現

点群 $\mathcal{P} = \{\mathbf{p}_i\}_{i=1}^N$ をトークン列 $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N\times C}$ として表し Transformer へ入力する．集合としての順序不変性を保つため，点インデックスに依存する位置埋め込みは用いず [14]，Self-Attention と点ごとの MLP により Permutation-equivariant な set-to-set 写像を実現する．

3.2　DiT ブロック

本モデルは L 層の DiT ブロックからなり，Self-Attention で点群内の関係を学習し，Cross-Attention で画像パッチ特徴を参照して観測輪郭や局所部位への整合を促す．非線形変換として MLP を併用し，点ごとの特徴更新を行う．

3.3　時間・画像条件の注入

時刻 t と区間長 dt を埋め込み $e_t(t), e_{dt}(dt) \in \mathbb{R}^H$ とし，画像グローバル特徴から得た $e_{\mathrm{img}} \in \mathbb{R}^H$ と加算して

$$\mathbf{c} = e_t(t) + e_{dt}(dt) + e_{\mathrm{img}} \tag{7}$$

をブロック共通の条件ベクトルとして用いる．各ブロックでは AdaLN-Zero [16] により $\mathbf{c}$ で特徴を変調し，学習初期の安定性を高める．dt を明示的に与えることで，区間平均速度場 $u(x_t, r, t)$ の推定に必要な更新幅情報をネットワークへ供給する．

3.4　DINOv3 特徴と Post-MHSA Adapter (PMA)

画像特徴抽出には凍結した DINOv3 [18] を用い，グローバル特徴 $\mathbf{z}_{\mathrm{img}}$ とパッチ系列特徴 Z_{ctx} を得る．$\mathbf{z}_{\mathrm{img}}$ は線形射影して e_{img} として AdaLN 条件へ加え，Z_{ctx} は Cross-Attention のコンテキストとして用いる（次元は H へ射影）．また Z_{ctx} に対し，MHSA+MLP からなる軽量な残差アダプタ (PMA) を一度だけ

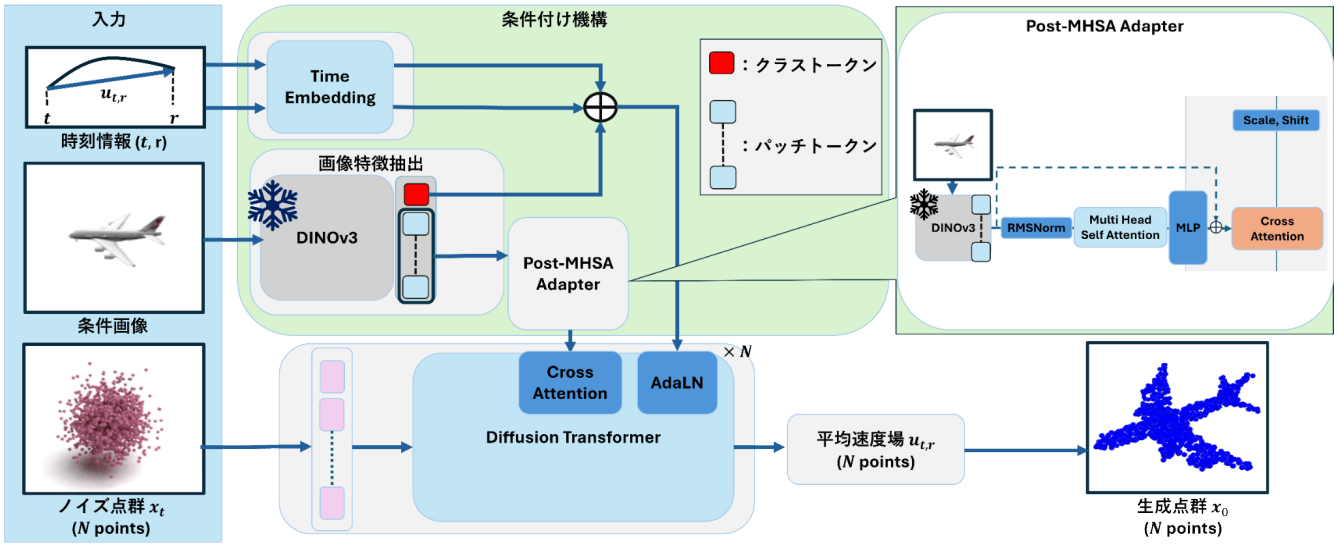

図 2　提案手法の全体構成

適用し，タスクに不要なばらつきを抑えつつ重要領域を強調したコンテキスト表現へ変換する．アダプタの出力射影はゼロ初期化し，学習初期の挙動を安定化する．

4. 損 失 関 数

4.1 CFG 付き平均速度場の導出

単一画像条件 c に整合する生成を強めるため，Mean Flow の平均速度場に Classifier-Free Guidance（CFG）[19] を導入し，サンプリング時のネットワーク評価回数（NFE）を増やさずに 1-NFE のままガイダンスを効かせる [4].

時刻 t における条件付き瞬間速度場を $v(x_t, t \mid c)$，無条件瞬間速度場を $v(x_t, t)$ とする．CFG によって条件付けられた瞬間速度は次で定義する.

$$v_{\mathrm{cfg}}(x_t, t \mid c) \coloneqq \omega\, v(x_t, t \mid c) + (1-\omega)\, v(x_t, t) \tag{8}$$

ここで ω はガイダンス強度であり，ω が大きいほど条件（cond）の寄与が強くなる.

この CFG 瞬間速度を区間 $0 \le r < t \le 1$ で平均した CFG 平均速度場を次で定義する.

$$u_{\mathrm{cfg}}(x_t, r, t \mid c) \coloneqq \frac{1}{t-r} \int_r^t v_{\mathrm{cfg}}(x_\tau, \tau \mid c)\, d\tau \tag{9}$$

式 (9) の両辺に $(t-r)$ を掛け，r を定数（$dr/dt = 0$）として t で微分すると，次の恒等式が得られる：

$$u_{\mathrm{cfg}}(x_t, r, t \mid c) = v_{\mathrm{cfg}}(x_t, t \mid c) - (t-r) \frac{d}{dt} u_{\mathrm{cfg}}(x_t, r, t \mid c) \tag{10}$$

全微分は接ベクトル $\dot{x}_t^{\mathrm{gen}} \coloneqq \frac{dx_t}{dt}$ を用いて次の形に展開できる.

$$\frac{d}{dt} u_{\mathrm{cfg}}(x_t, r, t \mid c) = \dot{x}_t^{\mathrm{gen}}\, \partial_x u_{\mathrm{cfg}}(x_t, r, t \mid c) + \partial_t u_{\mathrm{cfg}}(x_t, r, t \mid c) \tag{11}$$

生成過程における真の接ベクトルは本来 $\dot{x}_t^{\mathrm{gen}} = v_{\mathrm{cfg}}(x_t, t \mid c)$ である．しかし $v(x_t, t \mid c)$ や $v(x_t, t)$ は周辺化を含み直接計算できないため，学習では観測可能な量で近似する．学習では独立線形パス (式 (3)) を用いて x_t を構成する．このサンプル速度（教師信号）は $v_t \coloneqq \epsilon - x_0$ で与えられる．本研究では，$v(x_t, t \mid c)$ の教師信号として v_t を用いる.

一方，無条件速度 $v(x_t, t)$ も直接得られないため，無条件側は条件ドロップで評価したネットワーク出力 $u_\theta(x_t, t, t)$ で近似する．また $r \to t$ の極限で平均速度は瞬間速度に一致するため，$u_{\mathrm{cfg}}(x_t, t, t) = v_{\mathrm{cfg}}(x_t, t)$ が成り立つ．これに合わせて，学習で用いる近似 CFG 瞬間速度（近似接ベクトル）$\tilde{v}_t$ を次で定義する.

$$\tilde{v}_t \coloneqq \omega\, v_t + \kappa\, u_\theta(x_t, t, t \mid c) + (1-\omega-\kappa)\, u_\theta(x_t, t, t) \tag{12}$$

ここで $\kappa \ge 0$ は条件付き出力も $\tilde{v}_t$ へ混ぜる係数であり，$\kappa = 0$ で標準の CFG 近似に戻る.

恒等式 (10) に現れる全微分の接ベクトル $\dot{x}_t^{\mathrm{gen}}$ を，学習では近似 $\tilde{v}_t$ で置き換える．これにより回帰ターゲットを次で定める：

$$u_{\mathrm{tgt}} \coloneqq \tilde{v}_t - (t-r) \frac{d}{dt} u_\theta(x_t, r, t \mid c) \tag{13}$$

全微分は Jacobian Vector Product (JVP) で一括計算できる.

$$\left(u_\theta, \frac{d}{dt} u_\theta\right) = \mathrm{jvp}(u_\theta,\ (x_t, r, t),\ (\tilde{v}_t, 0, 1)) \tag{14}$$

ターゲット側を stop-gradient で固定し，次の損失を最小化する.

$$\mathcal{L}_{\text{MF-CFG}}(\theta) = \mathbb{E}\left[\left\| u_\theta(x_t, r, t \mid c) - \mathrm{sg}(u_{\mathrm{tgt}}) \right\|_2^2\right] \tag{15}$$

$r = t$ では $(t-r) = 0$ となり，(15) は通常の Flow Matching に退化する.

4.2 Geometry Noise Anchor

Mean Flow では，区間平均速度場 $u(x_t, r, t)$ を用いて, 以下のように大きな区間更新を行う.

$$\hat{x}_r = x_t - (t-r)\, u_\theta(x_t, r, t \mid c) \tag{16}$$

しかし，$r \approx 0$（実データ側）では，(i) 速度場推定の誤差が $(t-r)$ 倍されて位置誤差として現れやすい．(ii) 点群は集合で

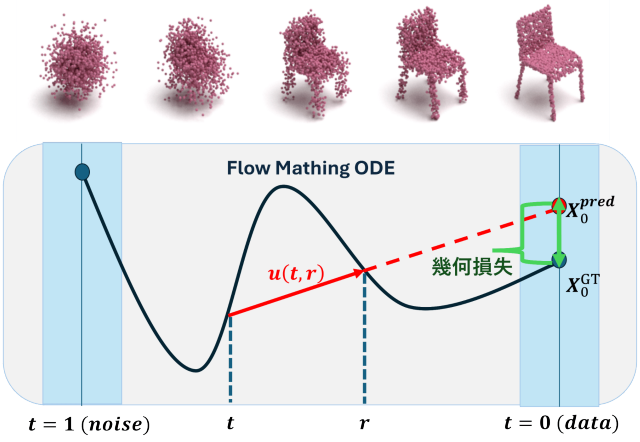

図 3　データ空間での補助損失

あり対応が自明でないため，点ごとの速度回帰だけでは表面への整列が弱い．(iii) 本手法の回帰ターゲットは自身のモデル出力を介した自己整合を含むため，データ近傍で生じた誤差が後続の更新にも伝播, 固定化されやすい．生成点群が表面からずれることがある．

そこで本研究では，平均速度場の学習損失に加えて，「データ空間の幾何整合」を直接拘束する補助損失 Geometry Noise Anchor (GNA) を導入する．本損失の要点は，時刻 r での予測速度場を, データ空間に外挿して損失を取ることで，データ付近の時刻での幾何的な整合性を高める点にある (図 3).

モデル出力 $u_\theta(x_t, r, t \mid c)$ から得られる時刻 0 での点群は式 (6) を用いると以下のように考えられる．

$$x_0^{\text{pred}} = x_t - (t-0)\, u_\theta(x_t, r, t \mid c) \tag{17}$$

Geometry Noise Anchor は，x_0^{pred} が x_0^{gt} に一致するよう，点群集合間距離 $D(\cdot,\cdot)$ を最小化する損失を導入する．

点群は順序を持たないため，D には Chamfer Distance (CD) のような set distance が自然である．本研究では Adaptive Probabilistic Matching Loss (APML) [20] を用いる．

$$\mathcal{D} = L_{\text{APML}}\left(x_0^{\text{pred}}, x_0^{\text{gt}}\right) \tag{18}$$

APML と主損失のスケール差を吸収するため，ミニバッチ平均を用いてスケール s を以下のように定義する．

$$s = \text{clip}\left(\frac{\mathbb{E}[\mathcal{L}_{\text{MF-CFG}}]_{\text{detach}}}{\mathbb{E}[\mathcal{L}_{\text{APML}}]_{\text{detach}} + \delta},\ s_{\min}, s_{\max}\right) \tag{19}$$

$s\,\mathcal{L}_{\text{APML}}$ としてオーダを揃える．また,t に依存する係数 $\lambda(t)$ を導入する．x_r^{pred} は区間幅 $(t-r)$ のジャンプにより得られるため，ステップ幅による正規化を導入し，$t \neq r$ の下で以下のような式にする．

$$\lambda(t) = \frac{\lambda_{\text{base}}}{\max(t-0, \tau)} \qquad (\tau > 0) \tag{20}$$

ここで τ は $(t-r) \to 0$ における発散を防ぐ下限である．また t が小さい場合に過度に重みが増大しないよう，τ により下限を設けて安定化する．λ_{base} は各時刻で，主損失に対して何割の幾何損失を入れるかのハイパーパラメータである．

最終的な総損失は以下のように定義する．

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MF-CFG}} + \lambda(t)\ s\ \mathcal{L}_{\text{APML}} \tag{21}$$

5. 実　　験

本章では，提案手法を評価するために行った実験について述べる本研究の目的は低 NFE で形状整合性の高い点群を生成すること, であるため，品質指標（CD/EMD/f-score）に加えて，推論時間と VRAM 使用量も併せて評価する．

5.1　データセット

本研究では単一画像条件付き点群生成の評価として ShapeNet [21] を用いた．ShapeNet は 55 カテゴリ・約 5.7 万個の 3D モデルを含む．

本研究では先行研究 [9], [11] に従い，各 3D モデルから点群をサンプリングして生成対象の点群 x_0 を構成する．学習・評価の安定化のため，点群は平行移動・スケーリングにより正規化する．入力画像は各モデルを複数視点からレンダリングした RGB 画像（1 オブジェクトにつき 24 視点）を用意し，学習時はランダムに視点を選択する．

5.2　評 価 指 標

本研究では，生成点群 $\hat{X}$ と正解点群 X の一致度を測るため，L2 Chamfer Distance と Earth Mover's Distance（EMD) を用いる．EMD は点数が一致していることを仮定するため，評価では両者の点数を同じ N に揃える．本研究で用いる Chamfer Distance は，二乗距離 $\|\cdot\|_2^2$ ではなく，L2 ノルム $\|\cdot\|_2$（平方根あり）に基づく対称形で定義する．評価実装では距離行列をユークリッド距離（平方根あり）で構成し，Hungarian 法により最適対応を求め，平均距離として EMD を算出する．CD は最近傍対応に基づくため高速に計算できる一方，多対 1 対応が起こり得るため密度偏りを評価しづらい場合がある．本研究では両者を併用することで，大まかな形状（CD）と全体対応の整合（EMD）を補完的に評価する．

F-Score は Melas-Kyriazi ら [22] での評価に基づき, 閾値 0.01 として評価する．なお, F-Score の評価では, 8192 点に RepKPU [23] を用いてアップサンプリングする．

本研究では低 NFE 生成の実用性を確認するため，推論時間と VRAM ピーク使用量を比較する．推論は単一 GPU（NVIDIA RTX A4000）上で実行し，ウォームアップ後に同一入力に対する推論を 100 回繰り返して平均と標準偏差を報告する．VRAM は推論 1 回あたりのピーク使用量を GiB 単位で報告する．いずれも画像特徴抽出から点群生成までのエンドツーエンドを対象とし，後処理は含めない．比較対象は点数 8192 の PC^2, 点数を 1024 に揃えた Transformer ベースの Point-E，RGB2Point，提案法（1-NFE) である．

5.3　実装の詳細

学習は A6000×8 の 8GPU 構成で実行した．点群は 1 サンプルあたり $N = 1024$ 点，画像入力は 224×224 の RGB 画像とした．事前学習は画像特徴抽出器（DINOv3）のみに用い，それ以外はスクラッチから学習した．DiT は隠れ次元 $D = 512$，ブ

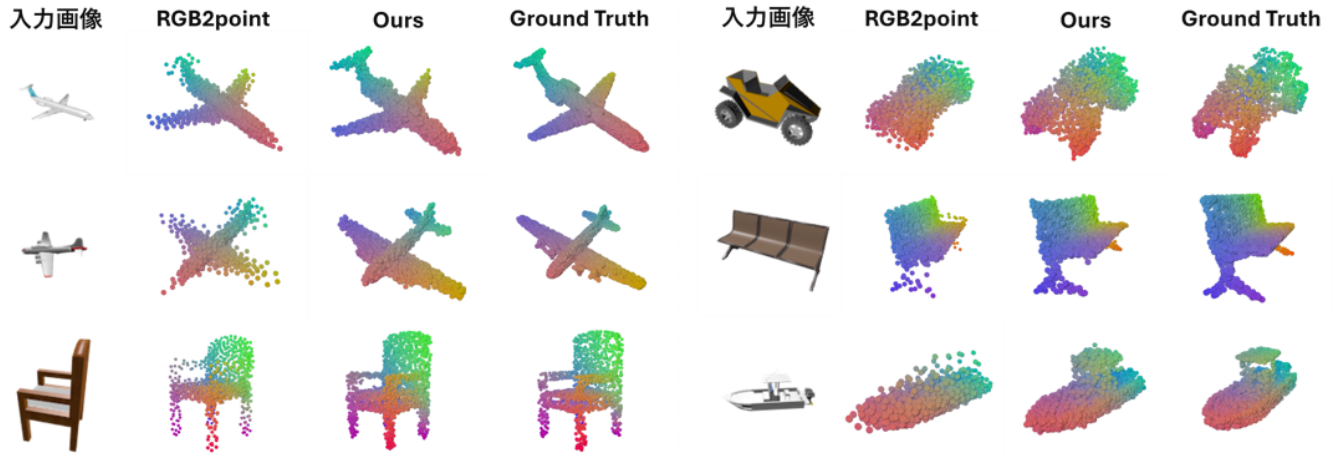

図 4　定 性 評 価

表 1　CD および EMD の比較，(CD×10^2，**太字**：最良値，下線：次点)

	CD×100 ↓				EMD×100 ↓			
Method	Car	Chair	Air	Mean	Car	Chair	Air	Mean
Self-Sup. [11]	5.48	10.91	7.11	7.11	4.95	14.93	11.07	10.31
DIFFER [10]	6.35	9.78	5.67	7.27	6.03	16.21	9.90	10.71
ULSP [12]	5.40	9.72	5.91	7.01	4.78	10.18	7.66	7.54
RGB2Point [13]	**4.22**	5.43	**2.70**	4.84	5.63	9.53	6.86	7.83
Ours	**4.22**	**4.29**	2.76	**4.51**	**3.82**	**4.29**	**3.04**	**4.13**
Ours w/o PMA	4.58	4.29	3.01	4.74	4.02	4.62	3.54	4.71
Ours w/o GNA	5.21	6.16	3.71	5.20	4.89	6.08	4.12	5.32

ロック数 $L = 12$，ヘッド数 $h = 8$ とし，画像条件には DINOv3（ViT-B）を用いた．また DINO のパッチトークンを処理する後段アダプタは 1 回のみ適用する設計とし，内部次元 1024，ヘッド数 4，出力射影はゼロ初期化とした．時刻サンプリングは logit-normal 分布に基づく設定を用いた．Flow Matching 成分と Mean Flow 成分は 50% ずつ混合した．Mean Flow 側では t と r を独立にサンプルし，`minmax` 方式により $t > r$ となるよう順序付ける．GNA に基づく幾何的拘束のハイパーパラメータである $\lambda_{\text{base}} = 0.4$ とした．

CFG における速度合成（学習で用いる近似 CFG 瞬間速度）は式 (12) で定義した．本実験では $(\omega, \kappa) = (1.0, 0.5)$ とした．学習時はラベルドロップアウト率を 0.1 とした．最適化には AdamW を用い，学習率は 1.0×10^{-4} とした．warmup 10,000 step を含むスケジューラを用い，バッチサイズ 128，総学習ステップ 120,000 step で学習した．

推論では平均速度場 $u_\theta(x_t, t, 0, c)$ を用いて $t \to 0$ の逆時間更新を行う．本研究では常に $r = 0$ を固定して，NFE=1 として式 (6) 用いる．

6. 実 験 結 果

6.1 定 量 評 価

表 1 に CD, EMD をまとめて示し，表 2 に F-Score の比較結果を示す．表 1 では，EMD が大幅に向上したことが分かる．これは，拡散モデルベースにおける，分布に着目した損失が上手く効いている．また，新規提案部分の有効性も確かめられた．

表 2 では，F-Score が既存手法を大きく上回っていることがわかる．特に，PC^2 [22] は 256-NFE の拡散モデルである．

表 3 に実行速度結果を示す．表 3 では，従来の拡散モデルベースよりも大幅に高速であり，フィードフォワード型モデルに匹敵する生成速度である．

表 2　閾値 0.01 における F-Score の比較 (**太字**：最良値，下線：次点)

Category	[9]	[24]	[22]	[13]	Ours
airplane	0.225	0.215	0.473	0.583	**0.591**
bench	0.198	0.241	0.305	0.406	**0.473**
cabinet	0.256	**0.308**	0.203	0.162	0.199
car	0.211	0.220	**0.359**	0.339	0.341
chair	0.194	0.217	**0.290**	0.195	0.249
display	0.196	0.261	0.232	0.235	**0.274**
lamp	0.186	0.220	**0.300**	0.211	0.276
loudspeaker	0.229	**0.286**	0.204	0.113	0.144
rifle	0.356	0.364	0.522	0.674	**0.691**
sofa	0.208	**0.260**	0.205	0.194	0.239
table	0.263	0.305	0.270	0.268	**0.309**
telephone	0.407	**0.575**	0.331	0.372	0.407
watercraft	0.240	0.283	0.324	0.406	**0.438**
Average	0.244	0.289	0.309	0.316	**0.353**

表 3　推論時間と VRAM ピーク使用量の比較（RTX A4000, 単一 GPU）

Method	Time [ms/sample]	Peak VRAM [GiB]
PC^2 [22]	48000 ± 200	1.730
Point-E [14]	8290 ± 10	1.270
RGB2Point [13]	18.39 ± 2.184	0.418
Ours (1-NFE)	29.45 ± 0.253	1.082

6.2 定 性 評 価

生成点群の形状再現性や点密度の偏りを視覚的に確認するため定性評価を行った (図 4)．

全体として，提案手法は点密度の偏りを抑えつつ，形状としての一貫性を保った点配置を生成できていた．この傾向は，

EMD が一貫して良い値を示した定量結果（表 1）とも整合的である．一方，Chamfer 距離を直接最小化する RGB2Point では，表面近傍への点の吸着が強く輪郭は鋭いが，局所的な密度偏りや特定部位への点の集中が生じる例が見られた．

7. ま と め

本研究では，低 NFE で形状整合性の高い点群生成を目的として，Mean Flow の平均速度場回帰を単一画像条件付き点群生成へ適用し，1-NFE のまま条件整合を強める CFG 付き学習（MF-CFG）と，データ近傍での幾何的整合を促す Geometry Noise Anchor（GNA）を提案した．ShapeNet による評価では，提案法は EMD において既存手法を上回り，点群全体としての被覆・密度整合に優れた生成が可能であることを示した．また推論時間の比較により，拡散系手法と比べて大幅に高速な推論を実現し，低 NFE 生成の実用性を確認した．

8. 課題と今後の展望

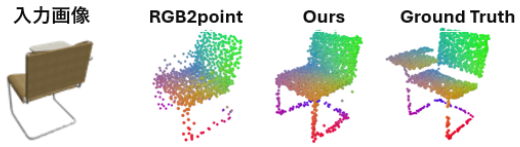

図 5 失 敗 例

失敗した生成例の 1 つを示す (図 5). 入力画像には映り込んでいない机を生成できていない. 単一画像のみを入力として扱うの場合, 尤もらしい背面形状を生成するのが限界であることがわかる. 柔軟な複数画像入力に対応したアーキテクチャの工夫が考えられる．

提案法の主損失 $\mathcal{L}_{\text{MF-CFG}}$ は Mean Flow 恒等式に基づく自己整合的ターゲットを含むため，学習時に JVP を計算する必要がある．JVP は追加の自動微分計算を伴い，学習時間および GPU メモリ使用量を増加させる．

また GNA では，点群間集合距離として APML を用いた．APML は $N \times N$ のコスト行列構成と Sinkhorn 反復を要し，$N = 1024$ でも CD に比べて計算・メモリ負荷が大きい．今後は，より複雑なデータセットでの検証をし，拘束を適用する時刻の調整などにより，学習コストを抑えつつ効果を維持する設計が課題である．

文 献

[1] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, et al., “3d gaussian splatting for real-time radiance field rendering.,” Acm Transactions on Graphics, Acm Transactions on Graphics, 2023.

[2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Advances in Neural Information Processing Systems, Advances in Neural Information Processing Systems, 2020.

[3] Y. Lipman, R.T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow matching for generative modeling,” Proc. of the International Conference on Learning Representations, 2023.

[4] Z. Geng, M. Deng, X. Bai, J.Z. Kolter, and K. He, “Mean flows for one-step generative modeling,” Advances in Neural Information Processing Systems, vol.38, 2025.

[5] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” Advances in Neural Information Processing Systems, Advances in Neural Information Processing Systems, 2017.

[6] W. Wu, Z. Qi, and L. Fuxin, “Pointconv: Deep convolutional networks on 3d point clouds,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.9621–9630, 2019.

[7] B. Graham, “Sparse 3d convolutional neural networks,” Proc. of The British Machine Vision Conference, pp.150–1, 2015.

[8] T. Li and K. He, “Back to basics: Let denoising generative models denoise,” arXiv:2511.13720, arXiv:2511.13720, 2025.

[9] C.B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, “3d-r2n2: A unified approach for single and multi-view 3d object reconstruction,” Proc. of the European Conference on Computer Vision, pp.628–644, 2016.

[10] K. L Navaneet, P. Mandikal, V. Jampani, and V. Babu, “Differ: Moving beyond 3d reconstruction with differentiable feature rendering,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[11] K. Navaneet, A. Mathew, S. Kashyap, W.-C. Hung, V. Jampani, and R.V. Babu, “From image collections to point clouds with self-supervised shape and pose networks,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.1132–1140, 2020.

[12] E. Insafutdinov and A. Dosovitskiy, “Unsupervised learning of shape and pose with differentiable point clouds,” Advances in Neural Information Processing Systems, vol.31, 2018.

[13] J.J. Lee and B. Benes, “Rgb2point: 3d point cloud generation from single rgb images,” Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp.2952–2962, 2025.

[14] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, “Point-e: A system for generating 3d point clouds from complex prompts,” arXiv:2212.08751, arXiv:2212.08751, 2022.

[15] X. Zeng, A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, and K. Kreis, “Lion: latent point diffusion models for 3d shape generation,” Advances in Neural Information Processing Systems, pp.10021–10039, 2022.

[16] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” Proc. of the IEEE/CVF International Conference on Computer Vision, pp.4195–4205, 2023.

[17] Y. Lan, S. Zhou, Z. Lyu, F. Hong, S. Yang, B. Dai, X. Pan, and C.C. Loy, “Gaussiananything: Interactive point cloud latent diffusion for 3d generation,” Proc. of the International Conference on Learning Representations, 2025.

[18] O. Siméoni, H.V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, et al., “Dinov3,” arXiv:2508.10104, arXiv:2508.10104, 2025.

[19] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications, 2021.

[20] S. Sharifipour, C.Á. Casado, M. Sabokrou, and M.B. López, “APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction,” Advances in Neural Information Processing Systems, vol.38, 2025.

[21] A.X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “ShapeNet: An Information-Rich 3D Model Repository,” arXiv:1512.03012, arXiv:1512.03012, 2015.

[22] L. Melas-Kyriazi, C. Rupprecht, and A. Vedaldi, “Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.12923–12932, 2023.

[23] Y. Rong, H. Zhou, K. Xia, C. Mei, J. Wang, and T. Lu, “Repkpu: Point cloud upsampling with kernel point representation and deformation,” Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.21050–21060, 2024.

[24] F. Yagubbayli, Y. Wang, A. Tonioni, and F. Tombari, “Legoformer: Transformers for block-by-block multi-view 3d reconstruction,” arXiv:2106.12102, arXiv:2106.12102, 2021.