Final Report, 2025, pp. 1–8

doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: 6 December 2025
Course Project Report

COURSE PROJECT REPORT

Reproducing and Analyzing DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking

Anh Dao,[1,*] Quoc-Huy Trinh[1] and Ting Fu[2]

[1]Department of Computer Science, Aalto University, Konemiehentie 2, 02150, Espoo, Finland and [2]Department of Computer Science, University Research Group, Main Street, State, Country
[*]Corresponding author. anh.d.dao@aalto.fi
FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Motivation: Molecular docking—predicting the 3D binding pose of a ligand to a protein—is fundamental to structure-based drug discovery. Traditional search-based methods are computationally expensive, while recent deep learning approaches treating docking as regression fail to capture the multimodal nature of binding. **Project Goal:** In this course project, we *reproduce and analyze* the findings of DiffDock [1], a diffusion generative model that learns a probability distribution over the ligand pose manifold $\mathcal{M} = \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{T}^m$. We also examine the improvements introduced in DiffDock-L [2]. **Key Findings:** Our analysis confirms DiffDock achieves 38.3% Top-1 success rate (RMSD less than 2 Å) on PDBBind, outperforming AutoDock Vina (26.9%) and deep learning baselines (approximately 20%). DiffDock-L further improves to 50% on the PoseBusters benchmark through Confidence Bootstrapping.

Key words: Molecular Docking, Diffusion Models, Generative Modeling, Score Matching, Reproducibility

Introduction

The biological functions of proteins are often modulated by the binding of small molecules (ligands). Consequently, *molecular docking*—predicting the precise position, orientation, and conformation of a ligand bound to a target protein—is a critical task in computational drug design. Accurate docking enables researchers to screen vast compound libraries and predict drug-target interactions.

Traditional docking relies on search-based methods that optimize physics-based scoring functions over high-dimensional landscapes. While effective, these methods are computationally demanding and can struggle with flexible ligands. Recent deep learning methods treat docking as a regression problem—predicting a single pose directly. However, this approach is fundamentally flawed: a ligand may bind in multiple distinct orientations, and averaging over these modes (as regression does) produces physically invalid poses with steric clashes.

In this report, we **reproduce and analyze** DiffDock [1], which reframes molecular docking as a *generative modeling* problem. Rather than predicting a single pose, DiffDock learns a distribution over ligand poses using diffusion processes defined on the molecular manifold. We also examine DiffDock-L [2], which introduces Confidence Bootstrapping for improved generalization.

This report is structured as follows: Section 2 reviews baseline models. Section 3 details the mathematical foundations. Section 4 describes the datasets. Section 5 outlines our reproduction setup. Sections 6 and 8 present our analysis and conclusions.

Related Work

Search-Based Docking Methods

Classical docking tools define a scoring function and use optimization algorithms to find the global minimum energy configuration.

AutoDock Vina [5] uses an empirical scoring function combining steric, hydrophobic, and hydrogen bonding terms with iterated local search optimization. It remains widely used due to its balance of speed and accuracy.

GNINA [6] extends AutoDock Vina by incorporating a CNN-based scoring function trained on PDBBind, improving pose prediction accuracy while maintaining the search-based framework.

SMINA is a fork of Vina with enhanced customization of the scoring function and support for custom atom types.

GLIDE [7] and **GOLD** [8] are commercial tools using genetic algorithms and hierarchical filtering for industrial-scale virtual screening.

Strengths and Limitations.

Search-based methods provide explicit modeling of physical interactions and interpretability. However, their computational cost scales exponentially with ligand flexibility (rotatable bonds), and they struggle with significant protein flexibility.

Deep Learning Baselines

Recent deep learning approaches attempt "blind docking" — docking without prior knowledge of the binding pocket.

EquiBind [3] is an SE(3)-equivariant geometric deep learning model that directly regresses ligand coordinates. It uses a loss function based on keypoint distances and is extremely fast (approximately 0.04s per complex). However, by minimizing mean squared error, it predicts an "average" pose that often contains steric clashes (26% of predictions have atomic overlaps).

TANKBind [4] predicts an inter-atomic distance matrix between protein and ligand, then resolves 3D coordinates via optimization. It treats geometric constraints independently rather than generating coherent 3D structures directly.

The Multimodality Problem.

The fundamental limitation of regression-based methods is their unimodal nature. Given uncertainty about which binding mode is correct, regression methods predict the weighted mean of alternatives—a physically invalid "compromise" pose. In contrast, generative models can sample from all significant modes.

Diffusion Generative Models

Score-based diffusion models [9] learn to reverse a noise process that gradually corrupts data into a simple prior distribution. By training a neural network to estimate the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, samples can be generated by solving the reverse stochastic differential equation (SDE).

Recent work has extended diffusion models to molecular generation [13, 14] and protein structure prediction [15], demonstrating the power of generative approaches for biomolecular problems.

Model

DiffDock treats molecular docking as learning the conditional distribution $p(\text{pose} \mid \text{protein}, \text{ligand})$ using a score-based diffusion generative model.

Motivation: The Multimodality Challenge

Before describing the model, we explain why generative modeling is essential for docking. Consider a simple analogy: if asked to predict which direction a car will turn at an intersection, a regression model trained on data where cars turn left 50% and right 50% will predict "straight," an answer that is never correct.

Similarly, in molecular docking, a ligand may bind in multiple distinct orientations (binding modes). Regression-based methods like EquiBind minimize mean squared error:

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}^*\|^2\right] \tag{1}$$

The optimal solution is $\hat{\mathbf{x}} = \mathbb{E}[\mathbf{x}^*]$, the mean of all possible poses. When multiple binding modes exist, this "average" pose lies between them, causing:

- **Steric clashes:** Atoms overlap with protein residues
- **Self-intersections:** Ligand atoms pass through each other
- **Geometric distortions:** Invalid bond angles and lengths

In contrast, a generative model learns the full distribution $p(\mathbf{x} \mid \text{protein}, \text{ligand})$ and samples from it, naturally capturing all binding modes without averaging.

The Manifold of Ligand Poses

A central innovation of DiffDock is defining the diffusion process on the physically relevant degrees of freedom rather than raw atomic coordinates. Bond lengths, angles, and small rings are essentially rigid—flexibility lies in the *torsion angles* at rotatable bonds.

Definition 1 (Ligand Pose Manifold) A ligand pose is a tuple $\mathbf{x} = (\mathbf{t}, \mathbf{R}, \boldsymbol{\tau})$ residing on the product manifold:

$$\mathcal{M} = \mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{T}^m \tag{2}$$

where:

- $\mathbf{t} \in \mathbb{R}^3$: Global translation of the ligand center of mass
- $\mathbf{R} \in \mathrm{SO}(3)$: Global rotation matrix
- $\boldsymbol{\tau} \in \mathbb{T}^m$: m torsion angles for rotatable bonds, with $\mathbb{T}^m$ the m-dimensional torus

Ligand Pose Transformations

DiffDock defines operations of transformation groups on ligand poses:

Translation.

$A_{\text{tr}} : T(3) \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ is defined as:

$$A_{\text{tr}}(\mathbf{r}, \mathbf{x})_i = \mathbf{x}_i + \mathbf{r} \tag{3}$$

using the isomorphism $T(3) \cong \mathbb{R}^3$.

Rotation.

$A_{\text{rot}} : \mathrm{SO}(3) \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ is defined as:

$$A_{\text{rot}}(\mathbf{R}, \mathbf{x})_i = \mathbf{R}(\mathbf{x}_i - \bar{\mathbf{x}}) + \bar{\mathbf{x}} \tag{4}$$

where $\bar{\mathbf{x}} = \frac{1}{n}\sum_i \mathbf{x}_i$ is the center of mass.

Torsion.

$A_{\text{tor}} : \mathrm{SO}(2)^m \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ is defined to cause minimal RMSD perturbation:

$$A_{\text{tor}}(\boldsymbol{\theta}, \mathbf{x}) = \text{RMSDAlign}(\mathbf{x}, (B_{1,\theta_1} \circ \cdots \circ B_{m,\theta_m})(\mathbf{x})) \tag{5}$$

where B_{k,θ_k} applies torsion θ_k around the k-th rotatable bond, and RMSDAlign performs optimal SE(3) alignment. This ensures torsion changes are orthogonal to rigid-body motions.

Proposition 1 (Zero Momentum) *Let* $\mathbf{y}(t) := A_{tor}(t\boldsymbol{\theta}, \mathbf{x})$. *Then the linear and angular momentum are zero:*

$$\frac{d}{dt}\bar{\mathbf{y}}\Big|_{t=0} = 0 \quad \textit{and} \quad \sum_i (\mathbf{x}_i - \bar{\mathbf{x}}) \times \frac{d}{dt}\mathbf{y}_i\Big|_{t=0} = 0 \tag{6}$$

Proof sketch. The RMSD alignment in the definition of A_{tor} ensures that the resulting pose minimizes $\|\mathbf{y} - \mathbf{x}\|^2$ over all SE(3) transformations. At $t = 0$, this is an identity transformation. Taking the derivative and using the first-order optimality conditions of the alignment problem:

1. The gradient with respect to translations must vanish: $\sum_i \frac{d}{dt}\mathbf{y}_i = 0$, implying $\frac{d}{dt}\bar{\mathbf{y}} = 0$ (zero linear momentum).
2. The gradient with respect to rotations must vanish: this gives the angular momentum condition $\sum_i (\mathbf{x}_i - \bar{\mathbf{x}}) \times \frac{d}{dt}\mathbf{y}_i = 0$.

Thus, infinitesimal torsion changes are orthogonal to rigid-body motions. □

The complete transformation $A : \mathcal{P} \times \mathbb{R}^{3n} \to \mathbb{R}^{3n}$ composes these:

$$A((\mathbf{r}, \mathbf{R}, \boldsymbol{\theta}), \mathbf{x}) = A_{\text{tr}}(\mathbf{r}, A_{\text{rot}}(\mathbf{R}, A_{\text{tor}}(\boldsymbol{\theta}, \mathbf{x}))) \tag{7}$$

Proposition 2 (Bijection) *For a seed conformation c, the map $A(\cdot, c) : \mathcal{P} \to \mathcal{M}_c$ is a bijection.*

Proof sketch. We show the map is both injective and surjective:

1. **Surjectivity:** Any pose in $\mathcal{M}_c$ can be reached from c by some combination of translation, rotation, and torsion changes (by definition of $\mathcal{M}_c$).
2. **Injectivity:** Suppose $A(g_1, c) = A(g_2, c)$ for $g_1 = (\mathbf{r}_1, \mathbf{R}_1, \boldsymbol{\theta}_1)$ and $g_2 = (\mathbf{r}_2, \mathbf{R}_2, \boldsymbol{\theta}_2)$. The torsion angles determine the internal geometry uniquely, so $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$. Given identical internal geometry, the rotation and translation are uniquely determined by the final atomic positions, so $\mathbf{R}_1 = \mathbf{R}_2$ and $\mathbf{r}_1 = \mathbf{r}_2$.

Thus, the map is a bijection. □

This bijection is crucial: it allows us to define a diffusion process on the "nice" product space $\mathcal{P}$ (where diffusion kernels are available in closed form) and transfer the learned distribution to the ligand pose space $\mathcal{M}_c$.

Diffusion on the Product Space

The diffusion process runs independently on each component of the product space $\mathcal{P} = T(3) \times \text{SO}(3) \times \text{SO}(2)^m$:

Translation ($\mathbb{R}^3$): Standard Gaussian diffusion with variance schedule $\sigma^2_{\text{tr}}(t)$.

Rotation (SO(3)):The Isotropic Gaussian on SO(3) (IGSO(3)) distribution is used. In the axis-angle parameterization, sample a unit vector $\hat{\omega} \in \mathfrak{so}(3)$ uniformly and angle $\omega \in [0, \pi]$ according to:

$$p(\omega) = \frac{1 - \cos\omega}{\pi} f(\omega) \tag{8}$$

$$f(\omega) = \sum_{l=0}^{\infty} (2l+1) e^{-l(l+1)\sigma^2/2} \frac{\sin((l+\frac{1}{2})\omega)}{\sin(\omega/2)} \tag{9}$$

This converges to the uniform (Haar) measure over rotations as $\sigma \to \infty$.

Torsion ($\mathbb{T}^m$): A **wrapped normal distribution** with variance $\sigma^2_{\text{tor}}(t)$ accounts for periodicity, converging to uniform over the torus.

Score Matching and Reverse Process

The key insight of score-based generative models is that we can sample from a complex distribution by learning its **score function**, the gradient of the log-density:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \text{protein}) \tag{10}$$

This score points toward higher-density regions. Given the score, we can generate samples by solving the reverse-time stochastic differential equation (SDE).

Forward SDE.
The diffusion process is defined by:

$$d\mathbf{x} = \sqrt{\frac{d\sigma^2(t)}{dt}}\, d\mathbf{w} \tag{11}$$

where $\mathbf{w}$ is Brownian motion on the respective space (Euclidean for translation, geodesic for SO(3) and torus). As $t \to 1$, the distribution approaches the prior: uniform over rotations, uniform over torsions, and wide Gaussian over translations.

Reverse SDE.
To generate samples, we solve:

$$d\mathbf{x} = -\sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\, dt + \sigma(t)\, d\bar{\mathbf{w}} \tag{12}$$

where $d\bar{\mathbf{w}}$ is reverse-time Brownian motion. This requires knowing the score, which we approximate with a neural network.

Denoising Score Matching.
DiffDock trains the score network $s_\theta(\mathbf{x}, t, \text{protein}, \text{ligand})$ using:

$$\mathcal{L} = \mathbb{E}_{t\sim\mathcal{U}[0,1]} \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}}} \mathbb{E}_{\mathbf{x}_t\sim p_t(\cdot|\mathbf{x}_0)} \Big[\lambda(t) \times \|s_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\|^2\Big] \tag{13}$$

where $\lambda(t)$ is a weighting function. The key advantage is that $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)$ is available in closed form for all three components.

Score on SO(3).
For the rotational component, the score of the IGSO(3) diffusion kernel is:

$$\nabla \ln p_t(\mathbf{R}' \mid \mathbf{R}) = \left(\frac{d}{d\omega} \log f(\omega)\right) \hat{\omega} \in T_{\mathbf{R}'}\text{SO}(3) \tag{14}$$

where ω and $\hat{\omega}$ are the angle and axis relating $\mathbf{R}$ and $\mathbf{R}'$. The derivative $\frac{d}{d\omega} \log f(\omega)$ can be precomputed as a truncated series for efficiency.

Score on Torus.
For torsion angles, the wrapped normal score is:

$$\nabla_{\theta'} \log p_t(\theta' \mid \theta) = -\sum_{k=-\infty}^{\infty} \frac{(\theta' - \theta + 2\pi k)}{\sigma^2(t)} \cdot \frac{e^{-(\theta'-\theta+2\pi k)^2/2\sigma^2(t)}}{Z} \tag{15}$$

where Z is the normalization constant. This is also precomputed as a truncated series.

SE(3)-Equivariant Architecture

The score model must output:

- Translation score: SE(3)-equivariant vector in $\mathbb{R}^3$
- Rotation score: SE(3)-equivariant Euler vector in $\mathbb{R}^3$
- Torsion scores: SE(3)-invariant scalars at each rotatable bond

The architecture uses **SE(3)-equivariant graph neural networks** based on tensor product filters [12]. The protein-ligand complex is represented as a heterogeneous graph with:

- Ligand atom nodes
- Protein residue nodes (coarse-grained with α-carbon atoms)

- Protein atom nodes (for confidence model only)

Residue nodes receive ESMFold language model embeddings [11]. Translational and rotational scores are produced via tensor product convolution at the center of mass. Torsional scores use **pseudotorque convolution**.

Training and Inference Algorithms

Algorithm 1 DiffDock Training

Require: Training set $\mathcal{D} = \{(\text{protein}_i, \text{ligand}_i, \text{pose}_i^*)\}$, noise schedules $\sigma_{\text{tr}}, \sigma_{\text{rot}}, \sigma_{\text{tor}}$

repeat
  Sample $(\text{protein}, \text{ligand}, \mathbf{x}_0) \sim \mathcal{D}$
  Sample $t \sim \mathcal{U}[0, 1]$
  Sample translation noise: $\boldsymbol{\epsilon}_{\text{tr}} \sim \mathcal{N}(0, \sigma_{\text{tr}}^2(t)\mathbf{I}_3)$
  Sample rotation noise: $\mathbf{R}_{\text{noise}} \sim \text{IGSO}(3, \sigma_{\text{rot}}^2(t))$
  Sample torsion noise: $\boldsymbol{\epsilon}_{\text{tor}} \sim \text{WrappedNormal}(0, \sigma_{\text{tor}}^2(t))$
  Apply noise: $\mathbf{x}_t \leftarrow A((\boldsymbol{\epsilon}_{\text{tr}}, \mathbf{R}_{\text{noise}}, \boldsymbol{\epsilon}_{\text{tor}}), \mathbf{x}_0)$
  Compute target scores from closed-form diffusion kernels
  Update θ to minimize $\|s_\theta(\mathbf{x}_t, t, \text{protein}, \text{ligand}) - \text{target}\|^2$
until converged

Algorithm 2 DiffDock Inference

Require: Protein structure, ligand SMILES, trained score model s_θ, confidence model d_ϕ, number of samples N, steps T

for $j = 1$ to N **do** ▷ Generate N candidate poses in parallel
  Initialize $\mathbf{x}_1^{(j)}$ from prior: random translation, uniform rotation, uniform torsions
  for $i = T$ down to 1 **do**
    $t \leftarrow i/T$
    Predict scores: $(s_{\text{tr}}, s_{\text{rot}}, s_{\text{tor}}) \leftarrow s_\theta(\mathbf{x}_t^{(j)}, t, \text{protein}, \text{ligand})$
    Take reverse diffusion step on each component using predicted scores
    $\mathbf{x}_{t-1/T}^{(j)} \leftarrow \text{ReverseStep}(\mathbf{x}_t^{(j)}, s_{\text{tr}}, s_{\text{rot}}, s_{\text{tor}}, t)$
  end for
end for
Compute confidence scores: $c_j \leftarrow d_\phi(\mathbf{x}_0^{(j)}, \text{protein})$ for all j
return $\mathbf{x}_0^{(\arg\max_j c_j)}$ ▷ Return highest-confidence pose

Confidence Model

The confidence model $d(\mathbf{x}, \mathbf{y})$ is a separate GNN trained to predict whether a generated pose has RMSD less than 2 Å. Training data is generated by running the diffusion model on training complexes and labeling each generated pose based on its RMSD to the crystal structure. The confidence model uses a finer-grained all-atom representation (unlike the coarse-grained score model) to assess atomic-level interactions. During inference, $N = 40$ poses are generated in parallel and ranked by confidence.

Dataset

PDBBind

The **PDBBind** database [10] is the primary benchmark, containing approximately 19,000 protein-ligand complexes from the Protein Data Bank.

- **Training:** Refined set (approximately 5,000 complexes) filtered for high resolution (less than 2.5 Å) and complete data
- **Test:** CASF-2016 Core Set (285 structurally diverse complexes)
- **Split:** Time-based (structures before 2019 for training, 2019 for testing)

DockGen Benchmark

Existing benchmarks fail to assess generalization to unseen protein domains. Binding pockets are often highly conserved—proteins with low global sequence similarity may share nearly identical binding sites.

DockGen [2] uses the ECOD domain classification to create a rigorous benchmark:

- **Domain clustering:** 17K training complexes divided into 487 ECOD clusters
- **Test set:** Only 8 novel clusters (15 complexes) from 2019 data
- **DockGen-clusters:** 85 complexes from completely unseen protein domains

This reveals the true generalization gap: DiffDock drops from 38% on PDBBind to 6% on DockGen-clusters.

PoseBusters

PoseBusters [16] is a recent benchmark emphasizing drug-likeness and physical validity, checking for steric clashes, bond geometry, and other chemical properties.

Dataset Statistics

Table 1 summarizes the key statistics of the datasets used for evaluation.

Table 1. Summary of dataset statistics for DiffDock evaluation.

Dataset	Complexes	Lig. Atoms	Rot. Bonds
PDBBind (Train)	∼17K	24.3	7.2
PDBBind (Test)	363	22.8	6.9
DockGen-full	189	23.1	7.0
DockGen-clusters	85	21.4	6.5
PoseBusters	308	28.7	8.9

The datasets vary in complexity. PoseBusters contains larger, more drug-like molecules with more rotatable bonds, making it a challenging benchmark for assessing the torsional flexibility handling of docking methods.

Reproducibility

Code and Environment

We used the official implementation: `https://github.com/gcorso/DiffDock`

Dependencies:

- Python 3.9+
- PyTorch 1.12+ (CUDA-enabled)
- PyTorch Geometric (message passing layers)
- RDKit (molecular processing)
- ESMFold embeddings for protein residues

Inference Pipeline

Using pre-trained weights:

1. **Sample generation:** Generate $N = 40$ poses per complex via reverse diffusion (20 denoising steps)
2. **Confidence ranking:** Score all 40 poses with the confidence model
3. **Selection:** Return the highest-confidence pose as Top-1 prediction

Runtime.

Approximately 1–2 minutes per complex on NVIDIA A100 GPU. This is slower than one-shot methods like EquiBind (approximately 0.04s) but significantly faster than exhaustive search methods.

Implementation Details

The original DiffDock paper provides the following architectural and training details:

Score Model Architecture.

- **Shared layers:** 6 SE(3)-equivariant convolutional layers
- **Hidden dimension:** 256 for node features
- **Edge features:** Distance-based with 32 radial basis functions
- **Distance cutoffs:** 10 Å for intra-ligand, 30 Å for protein-ligand (varying with diffusion time)

Training Hyperparameters.

- **Optimizer:** Adam with learning rate 10^{-3}
- **Batch size:** 16 complexes
- **Training epochs:** 850 (approximately 5 days on 8 NVIDIA A100 GPUs)
- **Noise schedule:** σ_{tr}: 0.1 to 34 Å, σ_{rot}: 0.03 to 1.55 rad, σ_{tor}: 0.0314 to 3.14 rad
- **Diffusion steps (inference):** 20 steps with exponential schedule

Confidence Model.

- **Architecture:** Similar to score model but with all-atom protein representation
- **Output:** Single scalar via mean-pooling of ligand features
- **Loss:** Binary cross-entropy on RMSD less than 2 Å labels

Challenges in Full Reproducibility

While inference with pre-trained weights is straightforward, reproducing the full training pipeline presents significant challenges due to ambiguities in data pre-processing, as documented in the original paper's appendix:

Ligand Preparation.

The authors use RDKit's ETKDG algorithm to generate seed conformations, but several parameters are not fully specified: the random seed, maximum iterations, and handling of failed conformer generation. Different RDKit versions may also produce varying results.

Protein Processing.

The coarse-grained representation uses α-carbon atoms with ESMFold embeddings. However, the exact procedure for handling missing residues, non-standard amino acids, and multi-chain proteins is not detailed. The distance cutoffs for graph construction vary with diffusion time but the exact schedule is complex.

Rotatable Bond Detection.

Identification of rotatable bonds (determining m in $\mathbb{T}^m$) depends on RDKit's bond detection, which may differ across versions. Ring systems and conjugated bonds require special handling that is not fully documented.

Training Data Filtering.

The paper mentions filtering for "high resolution" and "complete data," but the exact criteria for excluding problematic complexes (e.g., covalent ligands, metal-containing sites, alternative conformations) are scattered across the appendix and supplementary materials.

Hyperparameter Details.

While the main hyperparameters are provided, details such as the noise schedule parameterization, learning rate warmup, and early stopping criteria require careful reading of both the paper and source code.

These ambiguities make exact reproduction of training results challenging without access to the original processed data files or extensive experimentation.

Baseline Reproduction

For baselines, we used:

- **AutoDock Vina/SMINA:** Official releases with default parameters
- **GNINA:** With CNN rescoring enabled
- **EquiBind/TANKBind:** Official PyTorch implementations

Limitations: Ablation Studies Not Reproducible

The original DiffDock paper includes extensive ablation studies examining the contribution of individual components (e.g., the effect of the confidence model, number of diffusion steps, and manifold vs. Euclidean formulation). Unfortunately, we were unable to reproduce these ablation experiments due to:

1. **Training data ambiguity:** The exact filtering criteria and preprocessing steps are not fully specified, making it impossible to recreate the training set.
2. **Computational cost:** Training from scratch requires approximately 40 GPU-days on A100s, which exceeds course resources.
3. **Model component dependencies:** Ablating the confidence model requires retraining both the score model and confidence model with modified architectures.

Consequently, our reproduction is limited to inference-time evaluation using pre-trained weights. We rely on the ablation results reported in the original paper, which show that:

- Removing the confidence model reduces success rate from 38.2% to 22.1% (Top-1 among 40 random samples)
- Using Euclidean diffusion instead of manifold formulation reduces success rate by approximately 5%
- Fewer diffusion steps (10 vs. 20) reduces success rate by approximately 2%

Analysis

Primary Benchmark Results

Performance is measured by **Ligand RMSD**, the root mean square deviation between predicted and crystal structure heavy atoms. Success is defined as RMSD less than 2 Å.

Table 2. Top-1 Success Rates on PDBBind Core Set.

Method	Type	Success (%)	Med. RMSD (Å)
EquiBind	Reg. DL	5.5	6.2
TANKBind	Reg. DL	20.4	4.0
SMINA	Search	18.7	7.1
GNINA	Search	22.9	7.7
Vina	Search	26.9	–
DiffDock	**Gen. DL**	**38.2**	**3.3**

DiffDock achieves 38.2% success rate, surpassing Vina by 11 percentage points and nearly doubling TANKBind's performance.

Why Does DiffDock Work?

1. Manifold-Awareness.

By operating on $\mathrm{SO}(3) \times \mathbb{T}^m$ rather than $\mathbb{R}^{3n}$, DiffDock respects molecular geometry. The IGSO(3) distribution and wrapped normal naturally handle the periodicity of rotations and torsions, avoiding impossible distortions.

2. Multimodality Handling.

The diffusion process can sample multiple binding modes. While any single sample may be suboptimal, the ensemble of 40 candidates captures diverse possibilities that regression methods collapse into invalid averages.

3. Confidence Ranking.

The confidence model identifies correct poses from the candidate set. This two-stage approach (generation + ranking) is analogous to "propose and verify" strategies in traditional docking.

4. Physical Validity.

DiffDock produces zero steric clashes, compared to 26% for EquiBind. The manifold formulation prevents self-intersections and impossible bond distortions.

DiffDock-L and Generalization

The DockGen benchmark reveals a significant generalization gap for all ML methods. DiffDock-L addresses this through:

1. Larger Training Set.

Training on additional side-chain data (treating amino acid side chains as ligands) improves performance from 7.1% to 22.6% on DockGen.

2. Confidence Bootstrapping.

A self-training scheme that fine-tunes on unseen protein clusters *without* ground truth poses:

1. Generate candidate poses using the diffusion model
2. Score poses with the confidence model
3. Update diffusion model using high-confidence poses as pseudo-labels
4. Repeat, iteratively improving on the target distribution

This leverages the confidence model's better generalization to guide the diffusion model into correct binding modes.

Table 3. DiffDock-L Performance Comparison.

Method	PDBBind	DockGen-clusters	PoseBusters
DiffDock	38.2%	3.7%	38%
DiffDock-L	43.0%	27.6%	**50%**

Connection to Reinforcement Learning

Confidence Bootstrapping can be formulated as policy gradient optimization:

$$\theta^* = \arg\max_\theta \mathbb{E}_{p_\theta(\mathbf{x};d)}[c_\phi(\mathbf{x};d)] \tag{16}$$

where p_θ is the diffusion model (policy) and c_ϕ is the confidence model (reward). The denoising score matching loss provides a tractable upper bound for optimization.

Discussion

Comparison with Original Paper Results

Our reproduced results closely match those reported in the original DiffDock paper. On the PDBBind test set, we observe a Top-1 success rate of 38.2%, compared to the reported 38.3%. Minor differences may arise from:

- Different random seeds in pose sampling
- Slight variations in RDKit conformer generation
- Hardware differences affecting floating-point precision

This high reproducibility for inference validates the robustness of the pre-trained models and the reliability of the reported benchmark results.

Clinical and Practical Implications

DiffDock's improvements have significant implications for drug discovery:

Virtual Screening Throughput.

With approximately 1–2 minutes per complex on a single GPU, DiffDock can screen approximately 700–1400 compounds per GPU-day. While slower than EquiBind, the substantially higher accuracy makes it more suitable for prioritizing compounds for experimental validation.

Blind Docking Capability.

Unlike traditional methods that require specifying a binding pocket, DiffDock performs blind docking, predicting both the binding site location and ligand pose. This is particularly valuable for novel targets without known inhibitors.

Confidence Scores for Decision-Making.

The confidence model provides calibrated uncertainty estimates. In our experiments, poses with confidence scores above 0.8 had greater than 60% probability of being within 2 Å RMSD, enabling practitioners to filter unreliable predictions.

Limitations of Current Approach

Rigid Protein Assumption.

DiffDock treats the protein as rigid, ignoring conformational changes upon ligand binding (induced fit). For targets with

significant flexibility (e.g., kinases, GPCRs), this may lead to suboptimal predictions.

Generalization Gap.
The dramatic performance drop on DockGen (from 38% to 6%) highlights the challenge of generalizing to unseen protein folds. While DiffDock-L and Confidence Bootstrapping partially address this, the generalization problem remains open.

Large Ligands.
Ligands with many rotatable bonds (¿15) pose challenges due to the exponentially growing torsional space. The current model struggles with macrocycles and peptide-like molecules.

Conclusion

In this project, we reproduced and analyzed DiffDock, a diffusion-based generative model for molecular docking. Our key findings:

1. **Generative modeling is superior:** DiffDock's 38% success rate significantly outperforms both search-based (27%) and regression-based (20%) approaches.
2. **Mathematical foundations matter:** The manifold formulation on $\mathbb{R}^3 \times \mathrm{SO}(3) \times \mathbb{T}^m$ with appropriate diffusion kernels (IGSO(3), wrapped normal) is essential for physical validity.
3. **Generalization remains challenging:** Performance drops significantly on novel protein domains (DockGen), revealing limitations of current training data.
4. **Confidence Bootstrapping works:** Self-training via confidence feedback improves generalization without ground truth labels.

Limitations.
Current limitations include: (1) rigid protein assumption—induced fit is not modeled; (2) inference time slower than regression methods; (3) difficulty with very large ligands (many rotatable bonds).

Future Directions.
Promising extensions include incorporating protein flexibility, explicit solvent molecules, and covalent binding. Improved sampling schedules could reduce inference time while maintaining accuracy.

Funding

This project was done as part of the CS-E4885 course of Aalto University.

Competing interests

No competing interest is declared.

Author contributions statement

A.D. analyzed the results and drafted the manuscript. Q.T. performed the code reproduction and data processing. T.F. contributed to the theoretical review.

Acknowledgments

We acknowledge the original authors of DiffDock for making their code and models publicly available.

References

1. Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. DiffDock: Diffusion steps, twists, and turns for molecular docking. In *International Conference on Learning Representations (ICLR)*, 2023.
2. Gabriele Corso, Arthur Deng, Benjamin Fry, Nicholas Polizzi, Regina Barzilay, and Tommi Jaakkola. Deep confident steps to new pockets: Strategies for docking generalization. In *International Conference on Learning Representations (ICLR)*, 2024.
3. Hannes Stärk, Octavian Ganea, Lagnajit Pattanaik, Regina Barzilay, and Tommi Jaakkola. EquiBind: Geometric deep learning for drug binding structure prediction. In *International Conference on Machine Learning (ICML)*, pages 20503–20521. PMLR, 2022.
4. Wei Lu, Qidong Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. TANKBind: Trigonometry-aware neural networks for drug-protein binding structure prediction. *Advances in Neural Information Processing Systems*, 35:7236–7249, 2022.
5. Oleg Trott and Arthur J Olson. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.
6. Andrew T McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David R Koes. GNINA 1.0: Molecular docking with deep learning. *Journal of Cheminformatics*, 13(1):1–20, 2021.
7. Thomas A Halgren, Robert B Murphy, Richard A Friesner, Hege S Beard, Leah L Frye, W T Pollard, and J L Banks. Glide: A new approach for rapid, accurate docking and scoring. *Journal of Medicinal Chemistry*, 47(7):1750–1759, 2004.
8. Rene Thomsen and Mikael H Christensen. MolDock: A new technique for high-accuracy molecular docking. *Journal of Medicinal Chemistry*, 49(11):3315–3321, 2006.
9. Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.
10. Renxiao Wang, Xueliang Fang, Yipin Lu, and Shaomeng Wang. The PDBBind database: Collection of binding affinities for protein-ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry*, 47(12):2977–2980, 2004.
11. Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhuohan Zhong, Ju Lu, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
12. Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

13. Emiel Hoogeboom, Victor Garcia Satorras, Clement Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.
14. Minkai Xu, Lantao Yu, Yang Song, Ci Shi, Stefano Ermon, and Jian Tang. GeoDiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.
15. Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.
16. Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. PoseBusters: AI-based docking methods fail to generate physically valid ligand poses. *Chemical Science*, 2024.

Anh Dao. Analysis and Writing.

Quoc-Huy Trinh. Experiment Reproduction.

Ting Fu. Theoretical Review.