FLOW-MATCHING BASED REFINER FOR MOLECULAR CONFORMER GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-energy molecular conformer generation (MCG) is a foundational yet challenging problem in drug discovery. Denoising-based generative models, including diffusion and flow matching, have become the dominant paradigm for MCG and achieved state-of-the-art performance by transporting samples from a simple base distribution to the molecular conformer distribution. However, sequential sampling can accumulate errors, especially in low-SNR regimes where inference states deviate from their timestep-conditioned training distributions. This train-inference mismatch makes vector-field prediction on off-trajectory states unreliable, further propagating errors through denoising. To address this issue, we propose a mismatch-aware flow-matching framework for molecular conformer refinement. Trained on data-centered perturbations, the refiner corrects upstream conformers without requiring explicit estimates of their residual error. Experiments on GEOM-QM9 and GEOM-Drugs show consistent quality improvements across strong denoising-based generators, while preserving diversity and reducing the total number of sampling steps.

1 INTRODUCTION

Low-energy 3D conformations, called *conformers*, determine a molecule's biological, chemical, and physical properties (Guimaraes et al., 2012; Schütt et al., 2018; Klicpera et al., 2019; Axelrod & Gomez-Bombarelli, 2020; Schütt et al., 2021). Therefore, generating accurate and diverse ensembles of conformers from the molecular graph is a fundamental task in computational chemistry. Traditional approaches can be grouped into two main categories: physics-based methods, such as molecular dynamics (Pracht et al., 2020), which explore conformational space with high fidelity at high computational cost; and cheminformatics methods (Hawkins et al., 2010; Riniker & Landrum, 2015), which are more efficient but less accurate, often trading precision for speed.

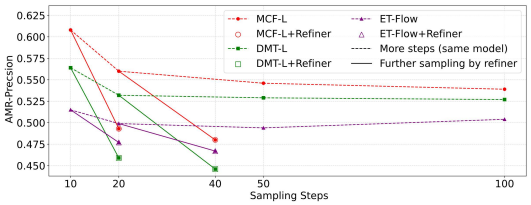

Figure 1: GEOM-DRUGS: increasing sampling steps vs. adding a refiner. Metric: Average Minimum RMSD (AMR)↓. The x-axis shows *total* sampling steps (including refiner steps). We compare allocating extra budget to adding our refiner (8.3M) to simply increasing the number of steps for three strong baselines: MCF-L (242M), ET-Flow (8.3M), and DMT-L (150M). For ET-Flow, the base architecture and predictor–corrector (PC) schedule are identical; only the refiner is added and fine-tuned.

Denoising-based generative models, including diffusion (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023; Tong et al., 2024), have developed rapidly in recent years. These methods address this gap in the MCG task by learning to sample from the distribution of low-energy conformers, balancing speed and accuracy. Existing work includes (1) methods based on pairwise atom distances in a distance matrix (Luo et al., 2021; Shi et al., 2021); however, they have too many degrees of freedom, which often leads to unstable optimization and poor results. (2) The method operates in torsional angle space (Jing et al., 2022); unfortunately, the performance is limited by the use of local information obtained by preprocessing tools such as RDKit. (3) More recent methods that predict 3D coordinates and report the current SOTA performance (Xu et al., 2022; Wang et al., 2024; Hassan et al., 2024; Liu et al., 2025). They work directly in 3D coordinate space to achieve the SOTA results, especially

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

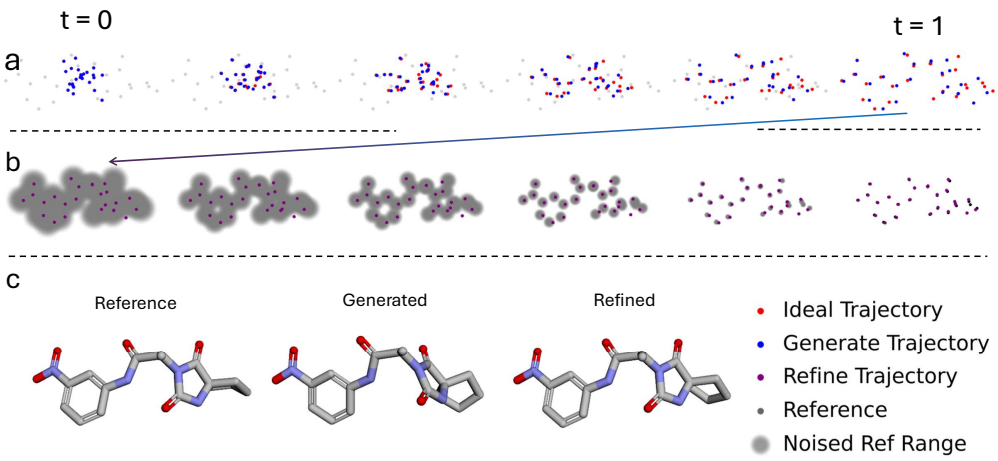

Figure 2: Overview of the Refiner pipeline. (a,b) PCA projections of 3D atomic coordinates. An upstream diffusion/flow-matching generator produces coarse conformers with accumulated geometric errors. The Refiner is trained on inputs that cover real-data geometries and corrects upstream conformers through a new refinement trajectory. (c) An example molecule showing improved geometry after refinement.

with large-scale models. However, methods based on 3D atom coordinates still suffer from issues introduced by the denoising method: models are hard to train to predict scores or vector fields for steps dominated by noise (Karras et al., 2022). Early diffusion steps that start from pure noise incur large errors. Because denoising is sequential and removes noise gradually, these early errors can accumulate and propagate through the trajectory, degrading final quality (Li & van der Schaar, 2024; Li et al., 2023; Chung et al., 2022). As errors accumulate, inference states may progressively deviate from the corresponding timestep-conditioned training distributions, becoming less compatible with their assigned time conditions and making subsequent predictions less reliable.

Generated conformers retain useful molecular structure, providing a natural, higher-SNR starting point for a new refinement trajectory. Starting from them avoids the pure-noise regime and its associated error accumulation, potentially improving final conformer quality. Yet their residual errors have unknown scale and form. Late-stage refiners such as the SD-XL refiner (Podell et al., 2023) specialize in predefined denoising steps and do not explicitly account for these unknown errors.

We propose a flow-matching refiner on data-centered perturbations and initialize a new refinement trajectory from each generated conformer. The scheduled perturbation scale decreases from a level chosen to cover upstream errors to zero. As refinement proceeds, the evolving conformer and the scheduled time condition can become sufficiently compatible to support effective correction, without explicitly estimating the input's error scale. We refer to this process as *self-calibration*. Limited early updates help preserve the input's molecular structure before effective correction begins.

Empirically, as shown in Fig. 1, simply adding steps improves only marginally, whereas our method achieves larger improvements with fewer additional steps. This advantage persists when the model architecture and PC sampler are held fixed. Our contributions can be summarized as follows:

1. We propose a mismatch-aware flow-matching refiner trained on data-centered perturbations to directly refine structured conformers from diverse upstream generators, without requiring prior knowledge of their residual error scales or structures.
2. We provide a mechanistic interpretation of self-calibration: the evolving conformer and scheduled time condition can become sufficiently compatible for effective correction. We analyze why early updates can remain controlled under time mismatch, helping preserve useful molecular structure.
3. The proposed pipeline improves conformer quality while preserving diversity and using fewer total sampling steps. These gains also hold in the controlled comparison with ET-Flow, where the refiner uses the same architecture and PC sampler as the upstream generator.
4. A single refiner improves conformers from the evaluated upstream generators without per-model tuning.

2 Related Work

Diffusion and Flow Matching. Diffusion models (Song et al., 2021; Ho et al., 2020) generate high-quality, diverse samples by learning the score of noise-perturbed data and numerically integrating a reverse-time SDE/ODE from a Gaussian prior to the data distribution. Flow Matching (FM) (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023; Liu et al., 2023) instead regresses a target vector field defined by stochastic interpolants, which also enables training with arbitrary source distributions (Pooladian et al., 2023). Recent work unifies diffusion and FM under stochastic interpolants, revealing them as closely related denoising generative models. (Albergo et al., 2023; Ma et al., 2024; Yu et al., 2024). However, in very low SNR regimes where noise dominates, the training targets become hard to learn (Karras et al., 2022; Hang et al., 2024), and during the early denoising steps, errors can accumulate and propagate, degrading sample quality (Li & van der Schaar, 2024; Chung et al., 2022; Li et al., 2023).

Refiners. Refinement modules are widely used in coarse-to-fine pipelines to recover high-frequency detail. Podell et al. (2023) introduces a diffusion-time–conditioned stage that polishes images, and Pandey et al. (2022) refines reconstructions produced by an upstream VAE. Restoration-style refiners target specific degradations such as deblurring and super-resolution (Whang et al., 2022; Saharia et al., 2022). However, existing refiners typically rely on a particular error type. Other methods repair outputs by injecting noise and then denoising (Kawar et al., 2022; Sawata et al., 2022); however, such noise-injection–based refiners must still start sampling by revisiting high-noise steps, thereby risking instability.

Molecular Conformer Generation. Deep learning methods for molecular conformer generation have attracted growing interest in recent years. Early attempts applied diffusion to distance matrices (Luo et al., 2021; Shi et al., 2021), but these approaches underperformed. Ganea et al. (2021) predict conformers via learned chemical structure parameters (e.g., bond lengths, bond angles, and torsions), yet overall quality is constrained by torsional accuracy; Jing et al. (2022) improves torsion prediction but remains limited by local structure. Subsequent works model and perturb directly in atomic coordinate space and scale to larger architectures (Xu et al., 2022; Wang et al., 2024; Hassan et al., 2024; Liu et al., 2025), achieving state-of-the-art performance. Nevertheless, sampling quality remains constrained by the intrinsic behavior of sequential denoising procedures.

3 Flow-Matching Based Refiner

Motivation. Current molecular conformer generation (MCG) methods predominantly rely on denoising generative models such as diffusion and flow matching, which achieve state-of-the-art performance (Hassan et al., 2024; Liu et al., 2025; Wang et al., 2024). However, early denoising steps starting from pure noise are difficult to learn (Karras et al., 2022). Since denoising is sequential, errors introduced in these highly noisy, low-SNR stages can propagate and accumulate throughout generation (Li & van der Schaar, 2024; Chung et al., 2022; Li et al., 2023). As a result, generated conformers retain meaningful molecular structure but contain residual errors.

Conformers generated by advanced models offer a different starting point. Because they already retain meaningful molecular structure, they typically lie in a higher-SNR regime than the pure-noise states used to initiate generation. Starting a new refinement trajectory from these conformers therefore bypasses the difficult-to-learn, low-SNR stages, avoiding the introduction and subsequent accumulation of errors from those stages while targeting the residual errors already present.

The challenge is that these residual errors have unknown scales and geometric forms that may vary across conformers and upstream generators. An SD-XL-style refiner (Podell et al., 2023) does not directly address this challenge: it specializes in prescribed late-stage noise levels, whereas it is difficult to determine which noise level, if any, appropriately characterizes the residual errors in a generated conformer. How can a refiner correct such errors without assuming a particular upstream error model or knowing each input's error scale in advance?

3.1 Refiner Definition

We model the residual error of an upstream-generated conformer relative to a reference conformer as

$$\widetilde{\mathbf{x}} = \mathbf{x}_1 + \mathbf{e}, \qquad \mathbf{e} = \sigma^\star \boldsymbol{\xi}, \tag{1}$$

Algorithm 1 REFINER (ODE)

Training

repeat
  sample $\mathbf{x}_1,\ \mathcal{G} = (\mathcal{V}, \mathcal{E}) \sim p_{\text{data}}$
  $\varepsilon \sim \mathcal{N}(\mathbf{0}, I_{3N}), \quad t \sim \text{Uniform}(0, 1)$
  $\mathbf{x}_0 \leftarrow \mathbf{x}_1 + \sigma\,\varepsilon$
  $\mathbf{x}_t \leftarrow (1-t)\,\mathbf{x}_0 + t\,\mathbf{x}_1 + s(t)\,\mathbf{z}$
  $\mathbf{u}_t \leftarrow -\,\sigma\varepsilon\ +\ s'(t)\,\mathbf{z}$
  Predict $\hat{\mathbf{u}} \leftarrow \mathbf{u}_\theta(\mathbf{x}_t, t, \mathcal{G})$
  $\mathcal{L} \leftarrow \|\hat{\mathbf{u}} - \mathbf{u}_t\|_2^2$; update $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
until convergence

Refinement

Require: Generated sample $\widetilde{\mathbf{x}}$ and its graph $\mathcal{G}$
  Trained refiner model $\mathbf{u}_\theta$
  Number of steps N
Schedule $\{t_n\}_{n=0}^{N}$ with $t_0 = 0, t_N = 1$
$\mathbf{x} \leftarrow \widetilde{\mathbf{x}}$
for $n = 0$ **to** $N-1$ **do**
  $\Delta t_n \leftarrow t_{n+1} - t_n$
  $\mathbf{x} \leftarrow \mathbf{x} + \Delta t_n \cdot \mathbf{u}_\theta(\mathbf{x}, t_n, \mathcal{G})$
end for
return $\mathbf{x}$

where $\mathbf{x}_1$ is a reference conformer, $\sigma^\star$ denotes the effective scale of the upstream error and $\boldsymbol{\xi}$ represents its unknown geometric form. Both the scale and geometric form may vary across conformers and upstream generators.

Error type. The geometric form of upstream errors is unknown and may vary across conformers and upstream generators. We therefore use isotropic Gaussian perturbations, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, I_{3N})$, to avoid favoring a generator-specific error pattern. This choice does not assume that upstream errors are Gaussian. Given zero mean and isotropic covariance, the Gaussian maximizes entropy (Cover & Thomas, 2006), making it a minimally biased choice for modeling perturbations with unknown geometric form.

Perturbation scale. The perturbation scale σ must balance two requirements. It should not be so large that noise obscures the molecular structure, returning refinement to the difficult-to-learn, low-SNR regime. Nor should it be so small that the training perturbations fail to cover the upstream errors we aim to correct. We therefore choose σ to cover the target error range ($\sigma > \sigma^\star$) while preserving meaningful molecular structure. Appendix A provides a reference estimate to guide the choice of σ.

During training, we construct the base distribution by perturbing data conformers:

$$\mathbf{x}_0 = \mathbf{x}_1 + \sigma\boldsymbol{\varepsilon} \tag{2}$$

Interpolant and target velocity. Given the training pair $(\mathbf{x}_0, \mathbf{x}_1)$ from Eq. 2, we define

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1 + s(t)\mathbf{z}, \qquad t \in [0, 1], \tag{3}$$

where $\mathbf{z}$ is an auxiliary noise variable sampled according to Hassan et al. (2024). Following their stochastic interpolant, we use $s(t) = \sqrt{t(1-t)}$. The corresponding target velocity for $t \in (0, 1)$ is

$$\mathbf{u}_t \ =\ \tfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_t = -\,\sigma\varepsilon\ +\ s'(t)\,\mathbf{z}. \tag{4}$$

Training objective. We train a time-conditioned vector field $\mathbf{u}_\theta(\mathbf{x}_t, t, \mathcal{G})$ to predict this target velocity:

$$\min_\theta\ \mathbb{E}\Big[\|\mathbf{u}_\theta(\mathbf{x}_t, t, \mathcal{G}) - \mathbf{u}_t\|_2^2\Big], \qquad t \sim \text{Unif}(0, 1). \tag{5}$$

As shown in Eq. 4, the target velocity is determined by the injected perturbations rather than explicitly by the reference coordinates $\mathbf{x}_1$, directing the learning objective toward perturbation correction.

Refinement of upstream conformers. At refinement time, we initialize the learned ODE with an upstream-generated conformer $\widetilde{\mathbf{x}}$:

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \mathbf{u}_\theta(\mathbf{x}_t, t, \mathcal{G}), \qquad \mathbf{x}_{t=0} = \widetilde{\mathbf{x}}. \tag{6}$$

Algorithm 1 summarizes the training and refinement process. Initializing an upstream conformer at $t = 0$ assigns it the training perturbation scale σ, even though its actual error scale $\sigma^\star$ is smaller. This raises two questions: when does the scheduled scale match the input's initial error scale, and how does the refiner behave before that matching time? We address these questions in Sections 3.2 and 3.3 respectively.

3.2 Matching the Upstream Error Scale

By construction, the training scale σ exceeds the effective error scale $\sigma^\star$ of the upstream conformers considered here. Along the training interpolant, the perturbation component $(1-t)\sigma\boldsymbol{\varepsilon}$ has scale

$$\sigma_t = (1-t)\sigma. \tag{7}$$

Because σ_t decreases continuously from σ to zero, there is a time $t^\star \in (0,1)$ at which it equals $\sigma^\star$:

$$(1-t^\star)\sigma = \sigma^\star, \qquad t^\star = 1 - \frac{\sigma^\star}{\sigma}. \tag{8}$$

During refinement, the model starts from an upstream conformer at $t=0$ and follows the decreasing scale schedule. At $t^\star$, the scheduled training scale matches the input's initial error scale $\sigma^\star$, without requiring an explicit estimate of that scale. This scale crossing motivates our interpretation of self-calibration: as refinement proceeds, the evolving conformer and its time condition can become sufficiently compatible to support effective correction. The effect of the additional stochastic interpolant term $s(t)\mathbf{z}$ on this scale analysis is discussed in Appendix B.

The upstream generation process ends at a nominally clean state, yet its output still contains accumulated errors. A new refinement trajectory provides a range of time conditions under which these residual errors can be corrected, without requiring their scale to be estimated in advance.

The time $t^\star$ characterizes scale matching for the fixed initial input. During refinement, however, the conformer's error scale evolves, so effective correction may begin at a different time. Section 3.3 discusses why early updates can remain controlled under time mismatch, helping preserve the input's useful structure before effective correction begins.

3.3 Refinement Behavior before Scale Matching

To understand the refiner's predictions under time mismatch, we consider two aspects: the correction behavior encouraged by its training target and whether upstream inputs retain geometric features encountered during training. We discuss the velocity target first, followed by the input representations, to explain why early updates can remain controlled despite the mismatched time condition.

Velocity under time mismatch. The objective in Eq. 5 trains the refiner to predict the velocity in Eq. 4, which contains only perturbation terms, with x_1 canceling at every time step. Thus, regardless of the assigned time, the learned task remains perturbation correction. Gaussian perturbations can also have small realized magnitudes at larger nominal scales, exposing the model to smaller deviations under these time conditions. This motivates correction behavior that can extend to upstream conformers despite time mismatch. Additionally, smaller errors may elicit smaller velocities, helping accurate conformers remain near their initial structures and preserve mode diversity.

Representation properties. We examine whether upstream conformers retain geometric features encountered during training, supporting the correction behavior described above under time mismatch. Following prior denoising-based approaches (Xu et al., 2022; Jing et al., 2022; Hassan et al., 2024), we use an SE(3)-equivariant model to predict $\mathbf{u}_\theta(\mathbf{x}_t, t, \mathcal{G})$. We focus on the relative distances, directions, and neighbor degrees used by the model.

Relative geometry under time mismatch. By SE(3)-equivariance, we may fix atom i at the origin through a global rigid transform. The local geometry is represented by relative vectors $r_{ij}(t) = \mathbf{x}_j(t) - \mathbf{x}_i(t)$ and the neighbor set $\mathcal{D}_R(i,t) = \{j \neq i \mid \|r_{ij}(t)\| \leq R\}$. We consider how time mismatch affects the distances and directions of these vectors, as well as the neighbor degree $|\mathcal{D}_R(i,t)|$. To analyze a pair of atoms, we approximate its relative vector by

$$r_{ij} = r_{ij}(1) + \sigma_{\text{eff}}\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I_3). \tag{9}$$

For independent isotropic perturbations of the two atoms, the relative-vector perturbation has effective scale $\sqrt{2}$ times the per-atom scale. For $t < t^\star$, the initial upstream error scale satisfies $\sigma^\star = (1-t^\star)\sigma < (1-t)\sigma$. Under this Gaussian approximation, the smaller-scale perturbations are concentrated within the region covered by the broader training perturbations: the latter expose the refiner to pairwise distances and directions around the reference geometry across a wider range of deviations (Anderson, 1958; Mardia & Jupp, 2009).This suggests that a smaller error scale need not

place the upstream input's pairwise geometry outside the range encountered during training at the assigned time t.

Neighbor degree under time mismatch. With a fixed radius cutoff R, the neighbor degree $|\mathcal{D}_R(i,t)|$ determines how many features are aggregated for atom i. Figure 3 shows a modest increase in mean degree as the training noise scale decreases, while the degree distributions remain largely overlapping across t. This overlap suggests that upstream conformers with smaller errors can retain familiar neighbor degrees, even if they have slightly more neighbors than typical training inputs at the assigned time t.

Together, the pairwise geometry analysis and the overlap in neighbor degree distributions suggest that upstream conformers retain features encountered during training at larger perturbation scales. This supports controlled correction updates despite time mismatch. Although these updates can change the error scale and shift the matching time from $t^\star$, limited cumulative displacement helps constrain error growth and preserve the upstream conformer's useful structure.

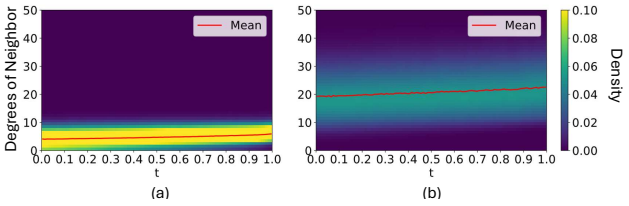

Figure 3: Neighbor degree distributions during training for radius cutoffs of (a) 2.5 and (b) 5.0.

4 Experimental Result

We evaluate the generator–refiner pipeline through three questions covering its effectiveness, refinement dynamics, and preservation of upstream conformational diversity.

RQ1: Effectiveness. Does the proposed pipeline improve overall conformer quality with fewer total sampling steps than generator-only sampling? Can the refiner trained with isotropic Gaussian perturbations also reduce different forms of geometric error? (Section 4.2)

RQ2: Refinement dynamics. Do the observed velocities and quality trajectories support controlled updates under time mismatch and effective refinement? (Section 4.3)

RQ3: Diversity preservation. Does refinement preserve the conformational diversity of upstream ensembles, as assessed through paired displacements and reference-conformer coverage? (Section 4.4)

4.1 Setup

Dataset. We evaluate on the GEOM dataset (Axelrod & Gomez-Bombarelli, 2022). GEOM-DRUGS is the largest relevant subset (304k drug-like molecules). We also train and evaluate on GEOM-QM9, a more established benchmark with smaller molecules. We follow Ganea et al. (2021) random splits of 80%/10%/10% into train/validation/test. Following Ganea et al. (2021); Jing et al. (2022), we use the same 1,000 random test molecules from the test set. The dataset splits are 106,586/13,323/1,000 (GEOM-QM9) and 243,473/30,433/1,000 (GEOM-DRUGS) molecules.

Implementation Details and Baselines. We implement our refiner by fine-tuning the open-source ET-Flow architecture (Hassan et al., 2024) and its weights. For fine-tuning, we reduce the learning rate from 0.007 to 0.001 and adopt a Cosine-Warmup learning rate schedule; otherwise, architectural choices and hyperparameters follow ET-Flow. During sampling, the refiner serves as a post-processor for conformers produced by the three strong denoising-based conformer generators: MCF (Wang et al., 2024), ET-Flow (Hassan et al., 2024), and DMT (Liu et al., 2025). For upstream sampling, we use the official open-source code implementations and trained weights. We compare the final performance with recent advanced models (Ganea et al., 2021; Xu et al., 2022; Jing et al., 2022; Wang et al., 2024; Hassan et al., 2024; Liu et al., 2025).

We set the perturbation scale to $\sigma = 1$ throughout, as it satisfies the two practical criteria discussed in Section 3.1 (coverage of plausible upstream errors and staying in a moderate-noise regime). We intentionally avoid dataset-specific tuning of σ to reduce overfitting and to keep the setting broadly

Table 1: Performance on the GEOM-QM9 test set, reported as Coverage (COV, %) and Average Minimum RMSD (AMR, Å). Since recent work already achieves 100% median COV at the commonly used threshold $\delta = 0.5$ Å and a median AMR below 0.05 Å, we adopt the more challenging COV threshold of $\delta =$ **0.05** Å. The best performance is **bold**.

	Recall				Precision			
	COV (**0.05**) ↑		AMR ↓		COV (**0.05**) ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
GeoDiff	-	-	0.297	0.229	-	-	1.524	0.510
GeoMol	-	-	0.225	0.193	-	-	0.270	0.241
Torsional Diff.	-	-	0.178	0.147	-	-	0.221	0.195
MCF-B (62M)	66.82	67.86	0.101	0.050	61.18	64.29	0.117	0.059
ET-Flow (8.3M)	75.72	87.23	0.083	0.031	70.32	75.00	0.114	0.053
DMT (55M)	72.90	83.33	0.087	0.036	67.76	75.00	0.107	0.047
MCF-B + **Refiner** (8.3M)	74.87	81.82	0.100	0.035	77.41	94.44	0.101	0.023
ET-Flow + **Refiner**	78.40	88.89	0.076	0.028	77.36	89.94	0.103	0.031
DMT-B + **Refiner**	**79.50**	**89.44**	**0.070**	**0.026**	**80.37**	**97.92**	**0.076**	**0.021**

Table 2: Performance on the GEOM-DRUGS test set, reported as Coverage (COV, %) and Average Minimum RMSD (AMR, Å). Coverage (COV) is computed with threshold $\delta = 0.75$ Å. The best performance is **bold**.

	Recall				Precision			
	COV ↑		AMR ↓		COV ↑		AMR ↓	
Method	Mean	Med	Mean	Med	Mean	Med	Mean	Med
GeoMol	44.60	41.40	0.875	0.834	43.00	36.40	0.928	0.841
GeoDiff	42.10	37.80	0.835	0.809	24.90	14.50	1.136	1.090
Torsional Diff.	72.70	80.00	0.582	0.565	55.20	56.90	0.778	0.729
MCF-L (242M)	85.10	92.86	0.390	0.343	66.63	70.00	0.623	0.546
ET-Flow (8.3M)	80.15	85.71	0.458	0.429	73.89	80.56	0.556	0.494
DMT-L (150M)	85.95	91.98	0.378	0.353	67.97	71.97	0.599	0.529
MCF-L + **Refiner** (8.3M)	86.44	93.68	0.368	0.330	72.07	78.41	0.550	0.480
ET-Flow + **Refiner**	80.29	85.11	0.439	0.410	74.58	**81.59**	0.530	0.467
DMT-L + **Refiner**	**87.47**	**94.12**	**0.349**	**0.319**	**75.91**	81.51	**0.497**	**0.446**

applicable. For reproducibility, we provide an anonymized implementation in the supplementary material.

Evaluation Metrics. Following Ganea et al. (2021); Jing et al. (2022), for each molecule with K reference conformers, we generate $2K$ candidate conformers. We report Average Minimum RMSD (AMR) precision, AMR recall, and Coverage (COV) (see Appendix C for details). Additional evaluations examine xTB-computed chemical properties (Appendix D) and per-conformer improvement and downgrade rates (Appendix E).

4.2 Refiner Effectiveness

Ensemble quality and sampling efficiency. We compare our generator–refiner pipeline with generator-only sampling under a smaller total sampling budget. The upstream baselines use 50 sampling steps, whereas our pipeline uses 40 steps: 20 for generation and 20 for refinement. Following the original papers, we use the Large (L) configurations of MCF and DMT on GEOM-DRUGS and their Basic (B) configurations on GEOM-QM9.

As shown in Tables 1 and 2, the proposed pipeline consistently improves precision AMR and COV across all three upstream generators on both datasets, despite using fewer total sampling steps. For example, refinement reduces the median precision AMR of DMT from 0.047 to 0.021 Å on GEOM-QM9 and from 0.529 to 0.446 Å on GEOM-DRUGS. Recall AMR also improves across all evaluated generators, while recall COV is generally maintained or improved. These results demonstrate that the refiner improves conformer quality without an overall loss of reference-conformer coverage.

Further experiments compare our approach with an SD-XL-style learned refiner and MMFF-based molecular mechanics optimization (Appendix F and G). We examine sensitivity to the training perturbation scale (Appendix H) and assess efficiency across sampling-step budgets and wall-clock runtime (Appendix I and J). Robustness on RDKit-generated conformers is evaluated in Appendix K.

Geometric error analysis. We train the refiner with isotropic Gaussian perturbations without assuming that upstream errors themselves are Gaussian. Under a fixed isotropic covariance, this maximum-entropy choice does not favor a particular perturbation direction. To examine whether the refiner can correct different forms of geometric error, we evaluate bond-length, bond-angle, and torsion-angle errors separately.

Table 3: Geometric errors on GEOM-DRUGS. Lower is better.

Method	Steps	Bond Length (Å)	Bond Angle (rad)	Torsion Angle (rad)
ET-Flow	50	0.069	0.012	0.023
ET-Flow + Refiner	20 + 20	**0.038**	**0.008**	**0.020**
Relative reduction	–	44.9%	33.3%	13.0%

As shown in Table 3, refinement reduces all three types of geometric error. This result shows that a refiner trained with isotropic Gaussian perturbations can correct different aspects of molecular geometry without explicitly modeling each error pattern.

4.3 Refinement Dynamics

We examine the refiner's sampling trajectory to assess whether its behavior is consistent with the analysis in Section 3.3. We use ET-Flow (Hassan et al., 2024) as the upstream generator and give the Refiner the same architecture, so that the observed changes can be attributed to refinement rather than an architectural difference. We focus on GEOM–QM9, where the relatively accurate upstream conformers make it a stricter test.

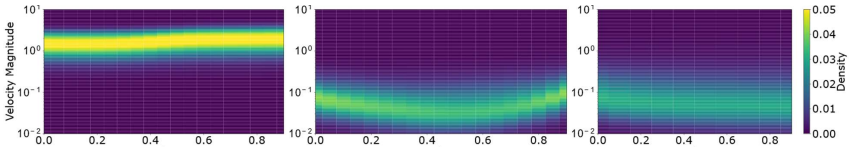

Figure 4: Velocity fields on GEOM–QM9: (a) ET–Flow sampling; (b) Refiner; (c) Refiner with randomized t.

Velocity dynamics. We first examine whether relatively accurate upstream conformers receive small predicted velocities, as suggested by the analysis in Section 3.3.

Figure 4(a) shows that ET-Flow typically predicts velocity magnitudes around 1 during generation. Figure 4(b) shows that the Refiner's velocities are concentrated below 0.1. Figure 4(c) further shows that these magnitudes remain similarly small when the Refiner's input time t is fully randomized. Thus, randomized time conditions do not substantially inflate the velocity scale, supporting limited update magnitudes under time mismatch. These observations are also consistent with the smaller-error, smaller-correction behavior discussed in Section 3.3. We next examine the cumulative effect of the scheduled refinement updates on conformer quality.

Two-stage refinement dynamics. Figure 5 shows two stages in the refinement trajectory. During the initial stage, $t \in [0, 0.3]$, RMSD increases slightly. This may partly reflect the neighbor-degree mismatch discussed in Section 3.3. Nevertheless, the limited increase supports controlled early updates despite time mismatch, consistent with the small predicted velocities shown in Figure 4(b)–(c).

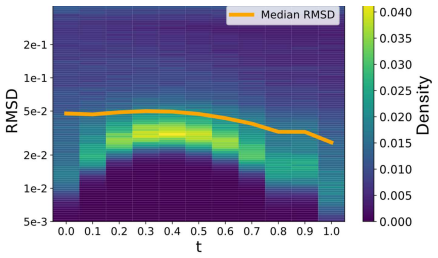

Figure 5: GEOM-QM9 AMR–precision dynamics during refinement.

In the later stage, RMSD decreases steadily, compatible with the self-calibration interpretation in Sec-

tion 3.2. Together, these dynamics support limited early error growth followed by effective refinement. By $t = 1$, median precision AMR decreases from approximately 0.05 Å to 0.03 Å, and a noticeable subset of conformers reaches RMSD below 0.01 Å. Additional training–test comparisons of geometric representations at $t = 0$ are provided in Appendix L.

4.4 Diversity Preservation

The recall results in Tables 1 and 2 show that refinement preserves the coverage of reference conformers. We now examine this behavior at the level of individual conformers. The small predicted velocities in Figure 4(b)–(c) suggest that the Refiner makes small updates to upstream conformers. To assess their cumulative effect, we measure the RMSD between each ET-Flow sample and its refined counterpart. For comparison, we also measure the RMSD between distinct reference conformers in each dataset.

Table 4: RMSD separation between distinct reference conformers and paired displacement from ET-Flow samples to their refined counterparts. All values are in Å.

Dataset	Comparison	Mean	Median
GEOM-QM9	Distinct reference conformers	0.963	0.968
	ET-Flow sample vs. refined counterpart	0.105	0.032
GEOM-DRUGS	Distinct reference conformers	2.056	2.029
	ET-Flow sample vs. refined counterpart	0.145	0.133

Table 4 shows that the paired displacement after refinement is much smaller than the typical separation between distinct reference conformers: approximately 9.2 times smaller by mean and 29.9 times smaller by median on GEOM-QM9, and 14.2 and 15.3 times smaller, respectively, on GEOM-DRUGS. This connects the small predicted velocities analyzed in Section 3.3 to the final conformers: their updates produce limited net displacement over the full refinement trajectory.

Relative to the separation between distinct conformers, such local movement makes transitions to different conformational basins less likely. Refinement can thus correct structural errors while retaining the conformational modes represented by the upstream ensemble, consistent with the preserved recall in Tables 1 and 2.

A finer-grained analysis of how refinement changes the coverage of individual reference conformers is provided in Appendix M.

5 Limitations

Our experiments focus on low-energy small-molecule conformers on GEOM-QM9 and GEOM-DRUGS, providing a controlled setting for evaluating refinement across upstream generators. Protein conformations and conditional generation involving binding pockets, intermolecular interactions, or solvent effects introduce different structural and interaction constraints that are not represented in these benchmarks. Adapting and evaluating the refiner in these settings remains future work.

6 Conclusion

We presented a flow-matching refiner that takes an upstream-generated conformer as its starting point. Because this conformer already contains meaningful molecular structure, refinement avoids restarting from pure noise and traversing the low-SNR stages of generation again. The refiner is trained on data-centered perturbations whose scale covers the residual errors of upstream conformers, allowing it to serve diverse generators without training a separate model for each one. Our analysis and experiments connect scale matching and local updates to improvements in conformer quality and the preservation of upstream diversity. The resulting generator–refiner pipeline achieves better performance with fewer total sampling steps.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

Generative AI tools were used to draft and edit the manuscript based on the authors' research ideas, analyses, and experimental results. The authors reviewed and revised all AI-assisted text and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

The main paper describes the refiner, its training and sampling procedures, the experimental settings, and the evaluation metrics. Additional experiments are provided in the appendix. An anonymized supplementary ZIP contains the code for reproduction.

REFERENCES

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.

T. W Anderson. *An introduction to multivariate statistical analysis*. Wiley, 1958.

Simon Axelrod and Rafael Gomez-Bombarelli. Molecular machine learning with conformer ensembles. *arXiv preprint arXiv:2012.08452*, 2020.

Simon Axelrod and Rafael Gomez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. Gfn2-xtb—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of Chemical Theory and Computation*, 15 (3):1652–1671, 2019.

Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Octavian Ganea, Lagnajit Pattanaik, Connor Coley, Regina Barzilay, Klavs Jensen, William Green, and Tommi Jaakkola. Geomol: Torsional geometric generation of molecular 3d conformer ensembles. *Advances in Neural Information Processing Systems*, 34:13757–13769, 2021.

Cristiano RW Guimaraes, Alan M Mathiowetz, Marina Shalaeva, Gilles Goetz, and Spiros Liras. Use of 3d properties to characterize beyond rule-of-5 property space for passive permeation. *Journal of chemical information and modeling*, 52(4):882–890, 2012.

Tiankai Hang, Shuyang Gu, Xin Geng, and Baining Guo. Improved noise schedule for diffusion training. *arXiv preprint arXiv:2407.03297*, 2024.

Majdi Hassan, Nikhil Shenoy, Jungyoon Lee, Hannes Stärk, Stephan Thaler, and Dominique Beaini. Et-flow: Equivariant flow-matching for molecular conformer generation. *Advances in Neural Information Processing Systems*, 37:128798–128824, 2024.

Paul CD Hawkins, A Geoffrey Skillman, Gregory L Warren, Benjamin A Ellingson, and Matthew T Stahl. Conformer generation with omega: algorithm and validation using high quality structures from the protein databank and cambridge structural database. *Journal of chemical information and modeling*, 50(4):572–584, 2010.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. *Advances in Neural Information Processing Systems*, 35:24240–24253, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *Advances in neural information processing systems*, 35:23593–23606, 2022.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations*, 2019.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023.

Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. In *ICLR*. OpenReview.net, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Zhiyuan Liu, Yanchen Luo, Han Huang, Enzhi Zhang, Sihang Li, Junfeng Fang, Yaorui Shi, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. Next-mol: 3d diffusion meets 1d language modeling for 3d molecule generation. In *ICLR*, 2025.

Shitong Luo, Chence Shi, Minkai Xu, and Jian Tang. Predicting molecular conformation via dynamic graph score matching. *Advances in Neural Information Processing Systems*, 34:19784–19795, 2021.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, 2024.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *arXiv preprint arXiv:2201.00308*, 2022.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings, 2023.

Philipp Pracht, Fabian Bohle, and Stefan Grimme. Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, 22(14):7169–7192, 2020.

Sereina Riniker and Gregory A Landrum. Better informed distance geometry: using what we know to improve conformation generation. *Journal of chemical information and modeling*, 55(12): 2562–2574, 2015.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

Ryosuke Sawata, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Takashi Shibuya, Shusuke Takahashi, and Yuki Mitsufuji. Diffiner: A versatile diffusion-based generative refiner for speech enhancement. *arXiv preprint arXiv:2210.17287*, 2022.

Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.

Kristof T Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.

Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pp. 9558–9568. PMLR, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*. OpenReview.net, 2021.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024, 2024.

Yuyang Wang, Ahmed A. A. Elhag, Navdeep Jaitly, Joshua M. Susskind, and Miguel Ángel Bautista. Swallowing the bitter pill: Simplified scalable conformer generation. In *ICML*, 2024.

Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16293–16303, 2022.

Edwin B. Wilson and Margaret M. Hilferty. The distribution of chi-square. *Proceedings of the National Academy of Sciences*, 17(12):684–688, 1931.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *ICLR*, 2022.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

APPENDIX

A RMSD UPPER BOUND WITH NOISE SCALE

After removing six SE(3) degrees of freedom, the non-rigid subspace has $d := 3N - 6$ degrees of freedom. we approximate the error Δ as isotropic Gaussian noise in coordinate space with unknown scale $\sigma^\star$, we can get:

$$\|\Delta\|^2 \;\sim\; \sigma^{\star 2}\,\chi^2_d, \tag{10}$$

where χ^2_d is a chi-square random variable with d degrees of freedom.

Consequently,

$$\mathrm{RMSD}(\widetilde{\mathbf{x}}, \mathbf{x}_1) \;=\; \sigma^\star \sqrt{\frac{1}{N}\chi^2_d} \tag{11}$$

By the Wilson–Hilferty approximation (Wilson & Hilferty, 1931), we have:

$$\left(\frac{X}{d}\right)^{1/3} \;\approx\; \mathcal{N}\!\left(1 - \frac{2}{9d}, \frac{2}{9d}\right) \quad \text{for } X \sim \chi^2_d. \tag{12}$$

by Eq.11, since $\mathrm{RMSD}^2 = (\sigma^{\star 2}/N)\, X$ with $X \sim \chi^2_d$,

$$\mathrm{RMSD} \;\approx\; \sigma^\star \sqrt{\frac{d}{N}\Big(1 - \frac{2}{9d} + z\sqrt{\frac{2}{9d}}\Big)^3}, \qquad z \sim \mathcal{N}(0,1). \tag{13}$$

Hence, denoting the standard-normal quantile at confidence level k by Q_k, the RMSD quantile at scale $\sigma^\star$ is:

$$\mathrm{RMSD}_{(Q_k,\sigma^\star)} \;\approx\; \sigma^\star \sqrt{\frac{d}{N}\left(1 - \frac{2}{9d} + Q_k\sqrt{\frac{2}{9d}}\right)^3}. \tag{14}$$

RMSD scale of the experimental perturbation. Section 4.1 uses $\sigma = 1$ in both datasets. To make the corresponding geometric scale explicit, Table 5 evaluates Eq. (14) at several atom counts. We use standard-normal quantiles $Q_{0.50} = 0$, $Q_{0.90} \approx 1.282$, $Q_{0.95} \approx 1.645$, and $Q_{0.99} \approx 2.326$.

Table 5: Approximate RMSD quantiles (in Å) of the isotropic training perturbation with $\sigma = 1$, computed from Eq. (14). Here N is the number of atoms and $d = 3N - 6$.

N	d	Median	90%	95%	99%
9	21	1.503	1.814	1.905	2.081
15	39	1.599	1.838	1.907	2.040
20	54	1.633	1.839	1.899	2.014
30	84	1.667	1.835	1.883	1.975
50	144	1.693	1.823	1.860	1.931

For these representative atom counts, 95% of isotropic perturbations have an approximate RMSD below 1.9 Å.

B EFFECT OF THE STOCHASTIC TERM ON SCALE MATCHING

The scale $\sigma_t = (1-t)\sigma$ in Eq. 7 describes the scale of the original perturbation term in the training interpolant. Including $s(t)\mathbf{z}$, the total perturbation is

$$(1-t)\sigma\boldsymbol{\varepsilon} + s(t)\mathbf{z}, \tag{15}$$

where $\boldsymbol{\varepsilon}$ and $\mathbf{z}$ are independent standard Gaussian variables. Its effective scale is therefore

$$\sigma_{\mathrm{eff}}(t) = \sqrt{(1-t)^2\sigma^2 + s(t)^2}. \tag{16}$$

At the nominal matching time $t^\star$ defined in Eq. 8,

$$\sigma_{\text{eff}}(t^\star) = \sqrt{(\sigma^\star)^2 + s(t^\star)^2} \geq \sigma^\star. \tag{17}$$

Since $s(1) = 0$, we have $\sigma_{\text{eff}}(1) = 0$. By continuity of $s(t)$, there exists $t^{\star'} \in [t^\star, 1)$ such that

$$\sigma_{\text{eff}}(t^{\star'}) = \sigma^\star. \tag{18}$$

Thus, the stochastic term may shift the matching time, but it does not eliminate scale matching along the training interpolant. In our setting, $\sigma = 1$ and $s(t) = \sqrt{t(1-t)}$, so $\sigma_{\text{eff}}(t) = \sqrt{1-t}$ remains strictly decreasing, and the matching time is $t^{\star'} = 1 - (\sigma^\star)^2$.

C Geometry Metric

Following Ganea et al. (2021); Jing et al. (2022); Hassan et al. (2024), the following works have used the so-called Average Minimum RMSD (AMR) and Coverage (COV) for Precision(P): Quality and Recall(R): the diversity, measured when generating twice as many conformers as provided by CREST. For $K = 2L$ let $\{C_l^*\}_{l\in[1,L]}$ for groundtruth and $\{C_k\}_{k\in[1,K]}$ for generated conformer.

$$\begin{aligned} \text{COV-R} &:= \frac{1}{L}\Bigg|\{l \in [1..L] : \exists k \in [1..K], \\ &\qquad RMSD(C_k, C_l^*) < \delta\Bigg| \\ \text{AMR-R} &:= \frac{1}{L}\sum_{l\in[1..L]} \min_{k\in[1..K]} RMSD(C_k, C_l^*) \end{aligned} \tag{19}$$

The δ is the coverage threshold, and the precision metrics are obtained by swapping ground truth and generated conformers.

D Chemical Property

Table 6: Median Boltzmann-weighted errors of ensemble properties between sampled and generated conformers: E, $\Delta\varepsilon$ in kcal/mol, μ in debye, and $E_{\min}$ in kcal/mol. Lower is better.

Method	E	μ	$\Delta\epsilon$	E_{min}
GeoDiff	0.31	0.35	0.89	0.39
GeoMol	0.42	0.34	0.59	0.40
Torsional Diff.	0.22	0.35	0.54	0.13
MCF-L	0.68	0.28	0.63	0.04
ET-Flow	0.23	0.19	**0.38**	0.02
MCF-L + **Refiner**	**0.20**	0.23	**0.38**	0.02
ET-Flow + **Refiner**	0.21	**0.18**	0.39	**0.01**

We also compare the chemical similarity between the generated and ground-truth conformers. We follow and use the same 100-molecule subset of Jing et al. (2022). For a molecule with K ground-truth conformers, we randomly select $\min(2K, 32)$ generated conformers, relax them with GFN2-xTB(Bannwarth et al., 2019), and compare Boltzmann-weighted ensemble properties between the generated and ground-truth sets. Using xTB(Bannwarth et al., 2019), we compute energy (E), dipole moment (μ), HOMO–LUMO gap ($\Delta\varepsilon$), and minimum energy ($E_{\min}$). Table 6 reports median errors, showing that our method can get better chemically accurate ensembles.

E Improvement and Downgrade Rates

Beyond the ensemble-level results in Section 4.2, we examine how refinement affects individual upstream conformers. For each conformer, we compare its precision RMSD with that of its refined

Table 7: Per-conformer improvement rate (IR) and downgrade rate (DR) (%) based on precision RMSD. A change must exceed the tolerance τ (Å) to be counted. Smaller thresholds are included for GEOM-QM9 because its upstream errors are smaller.

Dataset	τ	MCF + Refiner		ET-Flow + Refiner		DMT + Refiner	
		IR	DR	IR	DR	IR	DR
GEOM-QM9	0.02	69.2	15.3	29.2	5.3	50.7	5.7
	0.05	45.1	12.5	15.1	1.7	28.7	2.2
	0.10	29.0	9.8	7.7	0.4	14.5	0.6
	0.20	15.6	8.0	3.1	0.1	5.5	0.1
GEOM-DRUGS	0.05	37.5	16.7	35.3	5.6	22.7	7.5
	0.10	23.9	7.0	15.7	1.2	9.7	1.8
	0.20	9.6	1.4	2.6	0.1	2.1	0.2
	0.50	0.5	0.0	0.0	0.0	0.0	0.0

counterpart. We count an improvement when precision RMSD decreases by more than τ, and a downgrade when it increases by more than τ; changes within τ are counted as neither.

Table 7 shows that improvement rates exceed downgrade rates for every upstream model on both datasets at thresholds up to 0.20 Å. At the largest GEOM-DRUGS threshold of 0.50 Å, almost all changes fall within the tolerance. These paired results show that the ensemble-level quality gains are accompanied by more individual improvements than downgrades. Representative cases are shown in Appendix N.

F Comparison with SD-XL-style Refinement

Table 8: Comparison with an SD-XL-style refiner on QM9. Lower AMR is better.

Method	Steps	Recall-Mean	Recall-Med	PREC-Mean	PREC-Median
ET-Flow	50	0.083	0.031	0.114	0.053
ET-Flow + SD-XL refiner	20+20	0.083	0.029	0.113	0.051
ET-Flow + Ours	20+20	**0.076**	**0.028**	**0.103**	**0.031**

To examine whether the gains arise simply from introducing a second-stage model, we construct an SD-XL-style baseline with the same architecture and training settings as our Refiner, apart from the training interpolant and timestep range. The baseline refiner retains the upstream generator's noise-to-data interpolant and is fine-tuned on $t \in [0.8, 1]$. During inference, ET-Flow performs 20 sampling steps over $t \in [0, 0.8]$, and the resulting intermediate state is passed to the baseline refiner for another 20 steps over $t \in [0.8, 1]$. Both stages use the same predictor–corrector sampling strategy as our pipeline.

In contrast, our Refiner is trained using the data-centered perturbation distribution in Eq. 2. Our pipeline first completes the upstream generation trajectory in 20 steps, then initializes a new refinement trajectory from the generated conformer and performs 20 refinement steps. Thus, both pipelines use 20 upstream steps and 20 second-stage steps, but differ in the second stage's training distribution and its placement relative to the upstream trajectory.

Table 8 shows that the SD-XL-style baseline yields only marginal improvements over the 50-step ET-Flow generator, whereas our method achieves lower recall and precision AMR. Under these controlled settings, the results support the benefit of the proposed data-centered refinement design beyond adding a late-stage specialist to the original generation trajectory.

G Comparison to Classical Refinement Methods

Classical refinement methods, such as MMFF and UFF, are based on hand-crafted molecular mechanics force fields that approximate the potential energy surface using fixed functional forms and predefined parameter sets, rather than learning directly from the data distribution. While these

Table 9: Comparison with classical MMFF refinement on GEOM-QM9.

Method	Recall				Precision			
	COV↑		AMR↓		COV↑		AMR↓	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
ET-Flow	75.07%	87.50%	0.082	0.034	67.49%	73.07%	0.126	0.066
ET-Flow + MMFF	16.83%	0.00%	0.182	0.135	15.23%	0.00%	0.187	0.147
ET-Flow + Refiner	78.46%	88.89%	0.076	0.027	76.26%	85.50%	0.107	0.038

methods have been widely used for geometry optimization, their limited expressivity has made them increasingly insufficient for refining conformers generated by modern generative models.

To illustrate this limitation, we apply MMFF post-optimization to the conformers generated by ET-Flow on the GEOM-QM9 dataset. The results are summarized in Table 9. Contrary to the intended effect, MMFF refinement substantially degrades performance across all RMSD-based metrics. Both recall and precision drop sharply, and AMR errors increase, indicating that MMFF optimization often moves the generated conformers away from the target data manifold.

In contrast, the learned Refiner consistently improves all evaluation metrics when applied to the same ET-Flow outputs. This highlights a fundamental difference between classical force-field-based refinement and data-driven refinement. The Refiner is trained to correct errors with respect to the empirical conformer distribution, whereas MMFF optimizes a fixed approximate energy function that may be misaligned with the geometry produced by state-of-the-art generative models.

H Sensitivity to the Perturbation Scale

Table 10: Sensitivity analysis of the perturbation scale σ on QM9. Lower AMR is better.

σ	Recall-Mean	Recall-Med	PREC-Mean	PREC-Median
1.0	0.076	0.028	**0.103**	**0.031**
0.5	**0.075**	**0.025**	0.105	0.035
0.2	0.076	0.027	0.112	0.047
0 (w/o refiner)	0.083	0.031	0.114	0.053

The perturbation scale σ controls the effective refinement range of the refiner. Intuitively, σ should be sufficiently large to cover the deviation scale of upstream conformers, while remaining within a moderate-noise regime. To evaluate the sensitivity of the method to σ, we vary the perturbation scale during training and report the results in Table 10.

We observe that performance improves once σ becomes sufficiently large to cover the typical upstream error scale, and remains relatively stable afterward (e.g., $\sigma = 0.5$ and $\sigma = 1$). In contrast, a smaller perturbation scale (e.g., $\sigma = 0.2$) under-covers larger geometric deviations, leading to weaker refinement and degraded precision. These results suggest that σ mainly controls the effective error coverage of the refiner and does not require precise tuning.

I Additional Results: Refiner vs. More Sampling Steps

We present a detailed ablation comparing (i) increasing the number of steps in the upstream sampler and (ii) applying the refiner with an equal or smaller total number of steps. Concretely, we evaluate base models at 10/20/50/100 steps and contrast them with "k steps + refiner" (10+10, 20+20). The results show that increasing the sampling steps alone yields only limited improvement, whereas applying the refiner yields substantially larger gains at comparable or lower computational budgets. Results on GEOM-Drugs and GEOM-QM9 are reported in Tables 12 and 11, respectively.

Table 11: Performance of generated conformer ensembles on the GEOM-QM9 test set, reported as Coverage (COV, %) and Average Minimum RMSD (AMR, Å). Coverage (COV) is computed with threshold $\delta = 0.05$ Å. The best performance is **bold**.

		Recall				Precision			
		COV ↑		AMR ↓		COV ↑		AMR ↓	
Method	steps	Mean	Med	Mean	Med	Mean	Med	Mean	Med
MCF-B	10	51.60	50.00	0.115	0.067	43.57	34.00	0.144	0.094
MCF-B	20	62.13	62.50	0.108	0.056	55.73	50.00	0.124	0.068
MCF-B	50	66.83	67.86	0.101	0.050	61.18	64.29	0.117	0.059
MCF-B	100	68.90	75.00	0.099	0.047	64.00	69.12	0.112	0.052
MCF-B + Refine	10+10	74.32	80.00	**0.097**	0.037	76.97	94.23	**0.100**	0.025
MCF-B + Refine	20+20	**74.87**	**81.82**	0.101	**0.035**	**77.41**	**94.44**	0.101	**0.023**
DMT-B	10	69.64	77.78	0.089	0.041	61.37	66.29	0.118	0.062
DMT-B	20	71.71	80.00	0.088	0.038	66.21	75.00	0.109	0.048
DMT-B	50	72.90	83.33	0.087	0.036	67.76	75.00	0.107	0.047
DMT-B	100	73.75	83.33	0.085	0.036	68.39	75.00	0.103	0.047
DMT-B + Refine	10+10	**79.74**	88.89	**0.069**	0.028	79.32	93.75	0.079	0.025
DMT-B + Refine	20+20	79.50	**89.44**	0.070	**0.026**	**80.37**	**97.92**	**0.076**	**0.021**
ET-Flow	10	75.07	87.50	0.082	0.034	67.49	73.07	0.126	0.066
ET-Flow	20	75.65	85.71	0.083	0.033	69.40	75.00	0.121	0.056
ET-Flow	50	75.72	87.23	0.083	0.031	70.32	75.00	0.114	0.053
ET-Flow	100	76.55	87.50	0.077	0.030	70.67	79.75	0.115	0.047
ET-flow + Refine	10+10	**78.66**	**89.74**	**0.075**	**0.027**	76.73	87.50	0.106	0.037
ET-flow + Refine	20+20	78.40	88.89	0.076	0.028	**77.36**	**89.94**	**0.103**	**0.031**

J Runtime and Computational Cost Analysis

Following prior work, we measure computational cost using wall-clock sampling time under a fixed hardware and batch configuration. All methods are evaluated on a single NVIDIA A100 GPU with batch size 64, using 10 sampling steps and each method's recommended numerical precision. Average runtimes are reported in Table 13.

Under the same number of sampling steps, ET-Flow and the Refiner achieve the lowest wall-clock cost among the compared methods. This is expected, as the Refiner shares the same network architecture and predictor–corrector sampling strategy as ET-Flow, differing only in learned parameters and input coordinates. Accordingly, each refinement step incurs essentially the same computational cost as ET-Flow.

We therefore adopt the number of sampling steps for computational cost. Exact FLOP counts are not well-defined due to dynamically constructed molecular graphs, whose neighborhood structures vary across sampling steps and methods.

K Robustness to Large Structural Errors

To evaluate robustness under more systematic and larger structural errors, we further consider RDKit-generated conformers as upstream inputs to the Refiner. RDKit-generated conformers exhibit substantially larger RMSD errors relative to the GEOM reference conformers than the learned-generator outputs considered above, making them a challenging upstream input for the Refiner. This setting, therefore, serves as a strong stress test for the Refiner under severe initialization errors.

Table 14 reports the recall and precision metrics on the QM9 dataset. As expected, raw RDKit conformers yield extremely poor performance across all metrics, with near-zero recall and precision under COV evaluations. This confirms that RDKit-based initialization corresponds to a much more challenging error regime.

Table 12: Performance of generated conformer ensembles on the GEOM-DRUGS test set, reported as Coverage (COV, %) and Average Minimum RMSD (AMR, Å). Coverage (COV) is computed with threshold $\delta = 0.75$ Å. The best performance is **bold**.

		Recall				Precision			
		COV ↑		AMR ↓		COV ↑		AMR ↓	
Method	steps	Mean	Med	Mean	Med	Mean	Med	Mean	Med
MCF-L	10	84.52	92.86	0.444	0.408	65.63	68.75	0.668	0.608
MCF-L	20	85.82	93.18	0.400	0.369	66.43	69.54	0.637	0.560
MCF-L	50	85.10	92.86	0.390	0.343	66.63	70.00	0.623	0.546
MCF-L	100	85.06	92.12	0.388	0.348	66.70	70.45	0.621	0.539
MCF-L + Refiner	10 + 10	85.61	93.67	0.386	0.348	71.82	76.67	0.559	0.493
MCF-L + Refiner	20 + 20	**86.44**	**93.68**	**0.368**	**0.330**	**72.07**	**78.41**	**0.550**	**0.480**
DMT-L	10	85.82	91.99	0.391	0.364	66.62	70.50	0.625	0.564
DMT-L	20	85.58	92.44	0.384	0.358	67.48	71.65	0.608	0.532
DMT-L	50	85.95	91.98	0.378	0.353	67.97	71.97	0.599	0.529
DMT-L	100	85.80	92.30	0.375	0.346	67.90	72.50	0.598	0.527
DMT-L + Refiner	10 + 10	**87.66**	93.45	0.352	0.329	74.92	80.97	0.512	0.459
DMT-L + Refiner	20 + 20	87.47	**94.12**	**0.349**	**0.319**	**75.91**	**81.51**	**0.497**	**0.446**
ET-Flow	10	79.41	85.02	0.467	0.439	72.33	78.77	0.577	0.515
ET-Flow	20	79.28	84.74	0.467	0.437	73.34	80.00	0.562	0.499
ET-Flow	50	80.15	85.71	0.458	0.429	73.89	80.56	0.556	0.494
ET-Flow	100	79.78	84.15	0.462	0.436	73.70	80.00	0.561	0.504
ET-Flow + Refiner	10 + 10	80.20	**85.71**	0.445	0.411	73.47	81.25	0.547	0.477
ET-Flow + Refiner	20 + 20	**80.29**	85.11	**0.439**	**0.410**	**74.58**	**81.59**	**0.530**	**0.467**

Table 13: Wall-clock sampling time under a fixed hardware and batch configuration.

Method	Avg. Time (s)	Precision
ET-Flow / Refiner	0.03	FP32
MCF-L	0.07	FP32
DMT-L	0.04	FP16

Despite this difficulty, applying the Refiner leads to dramatic improvements across all evaluation metrics. Both recall and precision increase by more than an order of magnitude, demonstrating that the Refiner effectively corrects large, systematic geometric deviations rather than merely refining small perturbations around high-quality initial conformers.

L Training-Test Distribution Comparison of Geometric Representations

To examine potential mismatch at the start of refinement, we compare the training inputs x_0 with the test-time upstream conformers $\hat{x}$, both assigned the time condition $t = 0$. We examine three geometric statistics: bond-length deviations, bond-angle deviations measured as $1 - \cos(\Delta\theta)$, and neighbor degrees.

Figure 6 shows that the test-time bond-length and bond-angle deviations lie largely within the ranges observed during training, and the neighbor-degree distributions also overlap substantially. Together, these observations suggest that the initial time mismatch does not place the upstream inputs outside the training ranges of the examined geometric statistics, supporting the representation-based explanation in Section 3.3.

Table 14: Robustness evaluation on QM9 using RDKit-generated conformers as upstream inputs.

Method	Recall				Precision			
	COV		AMR		COV		AMR	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
RDKit	2.73%	0.00%	0.2313	0.1728	1.73%	0.00%	0.5120	0.4878
RDKit + Refiner	77.88%	87.50%	0.0780	0.0294	76.02%	83.33%	0.1071	0.0395

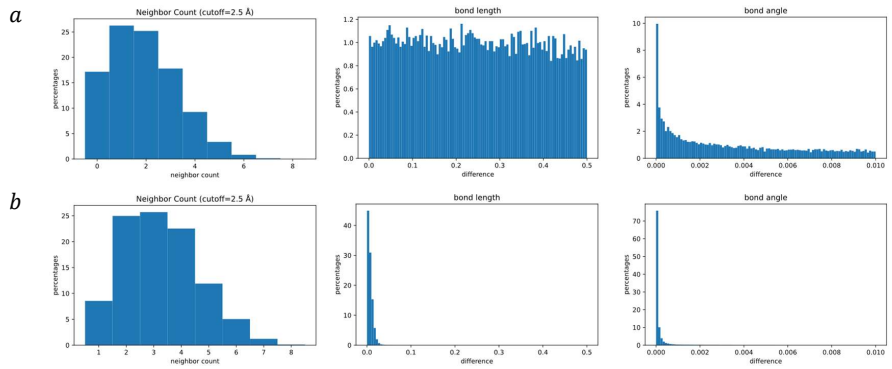

Figure 6: Comparison between (a) the training distribution of x_0 and (b) the test-time Refiner inputs $\hat{x}$. We visualize three geometric components: bond lengths, bond angles (measured as $1 - \cos(\Delta\theta)$), and neighbor degrees. The test-time distributions largely fall within the ranges observed during training for these geometric statistics, consistent with the analysis in Section 3.3.

M Recall Preservation Study

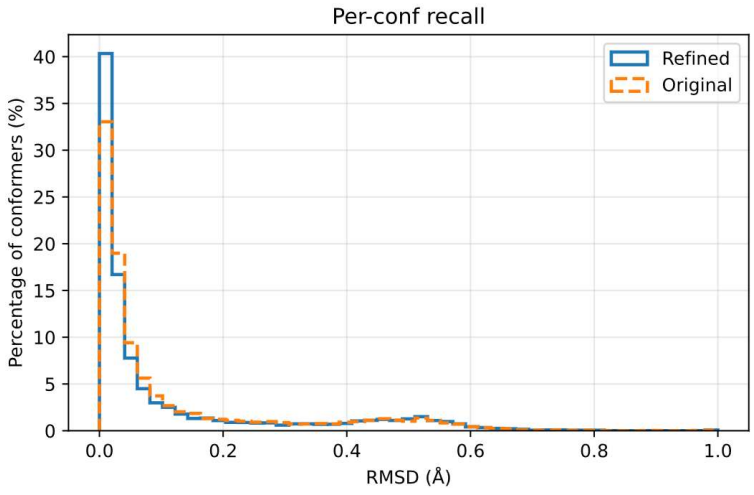

Figure 7: Per-conformer recall distribution before and after refinement, grouped by RMSD bins.

Figure 7 further supports this observation by examining recall improvements across different RMSD bins. The gains in recall are mainly concentrated in the low-RMSD regions, highlighting the Refiner's strength as a local geometric optimizer for already reasonable conformers. For ground-truth conformers with large RMSD-based recall gaps (i.e., those far from any generated conformer), the improvement is modest, and the curves remain nearly unchanged before and after refinement.

We observe no systematic deterioration in the reported reference-coverage statistics. Together with the small paired displacements in Table 4, these results support local refinement without an observed overall loss of reference coverage on the evaluated datasets.

N CASE STUDY

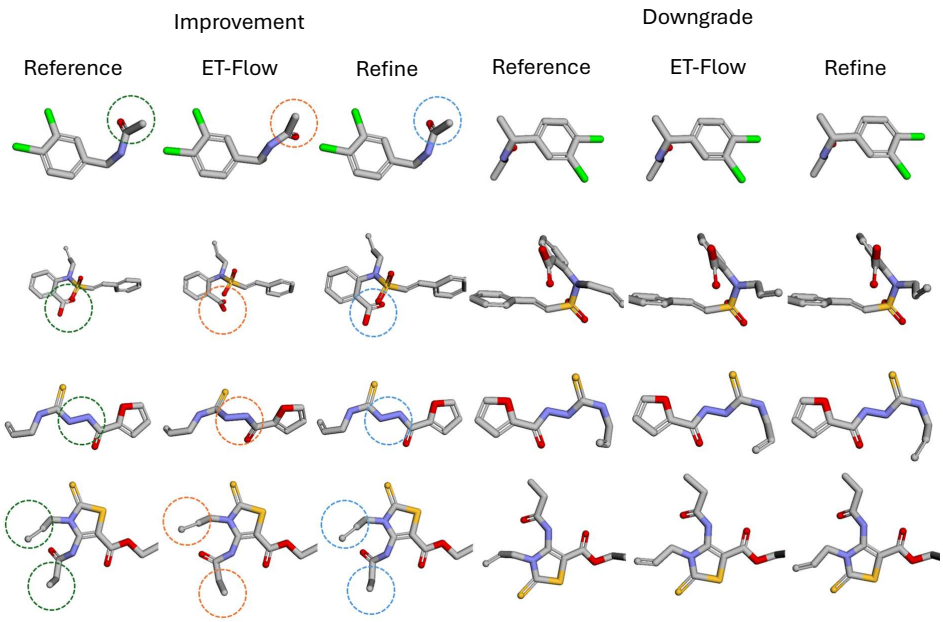

Figure 8: Qualitative visualization of the Refiner. The left column shows the conformers with the largest improvements, while the right column shows those with the largest downgrades. Each row shows reference, generated, and refined conformers of the same molecule. In the left column, the circles highlight the main structural differences among the reference, generated, and refined conformers.

We provide case studies to illustrate better how the Refiner behaves in practice. We first randomly sample molecules, and for each selected molecule, we identify the generated conformers that exhibit the largest improvement and the largest degradation in RMSD after refinement. For each such conformer, we visualize the reference, the upstream generated, and the refined structures. Within the same molecule, different generated conformers therefore serve as paired examples of substantial improvements and mild downgrades.

In the improvement cases (left column of Figure 8), we observe that the upstream ET-Flow samples retain certain structural errors, such as distorted local geometries, which the Refiner subsequently corrects. In contrast, for the downgrade cases (right column), the refined conformers remain broadly consistent with the corresponding generated ones: there is no evidence of new systematic errors being introduced. The observed degradations tend to be small and localized, and can be attributed to minor fluctuations in model capacity rather than to instability of the refinement process.

Taken together, these observations suggest that the Refiner acts as a local geometric corrector: for reasonably good upstream conformers, it can reliably improve local structure, which in turn yields consistent macroscopic gains in overall quality. This trend is quantified in Tables 7, which report clear net improvements despite a small number of downgraded conformers. The visualizations further reveal that the Refiner can repair structural issues (e.g., clearly mismatched torsional angles), whereas the downgraded cases remain local and do not introduce new structural errors.