SSD: Shell-Guided Spherical Diffusion for Molecular Geometry Generation

Yun-Yen Chuang[1,2] **Chen-Sheng Gu**[1,2] **Hung-Min Hsu**[3] **Kevin Lin**[4] **Ray-I Chang**[2]
[1]Maxora AI [2]National Taiwan University [3]University of Washington [4]Microsoft
{yunyenchuang, chenshenggu}@maxora.ai hmhsu@uw.edu keli@microsoft.com rayichang@ntu.edu.tw

Abstract

Diffusion models for 3D molecular geometry almost universally adopt an isotropic Gaussian prior, whose Frobenius-norm scale $\sigma_T\sqrt{3n}$ is governed by the noise level and atom count rather than by chemistry, so a per-atom scale mismatch with the data induces an absolute transport discrepancy that grows with $\sqrt{n}$, yielding high spatial entropy and unstable early trajectories. We introduce **Shell-guided Spherical Diffusion (SSD)**, a model-agnostic framework that replaces the Gaussian prior with a chemically scaled spherical-shell initialization, augments the forward process with radial attraction toward the shell, and, for the score-based backbones, samples with a geometry-guided hybrid reverse process that combines radial attraction, short-range repulsion, and an SE(3)-equivariant correction field. This joint design of initialization and dynamics is essential: neither a shell alone nor radial fields alone reproduces the stability or accuracy of SSD. We evaluate SSD across five representative coordinate-space backbones—GeoDiff, SubGDiff, EDM, SemlaFlow-style flow matching, and MCF—each under its canonical evaluation protocol. SSD consistently improves both quality and diversity under identical training and sampling budgets, upgrading weaker backbones such as GeoDiff and EDM to match or surpass stronger diffusion-, VAE-, and flow-based baselines. SSD therefore serves as a plug-in geometric enhancement that strengthens coordinate-space molecular generation models without modifying their architectures, loss functions, or training pipelines.

1 Introduction

Generating accurate 3D molecular structures is a core problem in computational chemistry, molecular design, and drug discovery [Hawkins, 2017, Kitchen et al., 2004, Schneider, 2018]. Classical conformer generators based on stochastic search or physics-based force fields often struggle with efficiency and scalability, especially for large or flexible molecules. Recent deep generative models alleviate these limitations by learning molecular geometry distributions directly from data [Mansimov et al., 2019, Gómez-Bombarelli et al., 2018, Walters and Barzilay, 2020, Polykovskiy et al., 2020], with diffusion-based coordinate generators achieving state-of-the-art performance on GEOM-QM9 and GEOM-Drugs [Ramakrishnan et al., 2014, Axelrod and Gomez-Bombarelli, 2022].

Diffusion-based coordinate models are used in two settings: *conditional generation*, which lifts a molecular graph into 3D [Xu et al., 2022, Shi et al., 2021, Zhang et al., 2024], and *unconditional refinement*, which denoises a noisy 3D geometry [Hoogeboom et al., 2022, Hua et al., 2024, Morehead and Cheng, 2024, Irwin et al., 2025, Dunn and Koes, 2024]. Despite architectural differences, nearly all such models adopt an *isotropic Gaussian prior* for initialization. By the Gaussian Annulus Theorem [Blum et al., 2020, Vershynin, 2018], a $3n$-dimensional Gaussian concentrates on a thin shell of Frobenius-norm radius $\sigma_T\sqrt{3n}$ (Appendix B), set by the noise level and atom count rather than by any chemical quantity. This mismatch with the molecule's own scale distorts score magnitudes

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

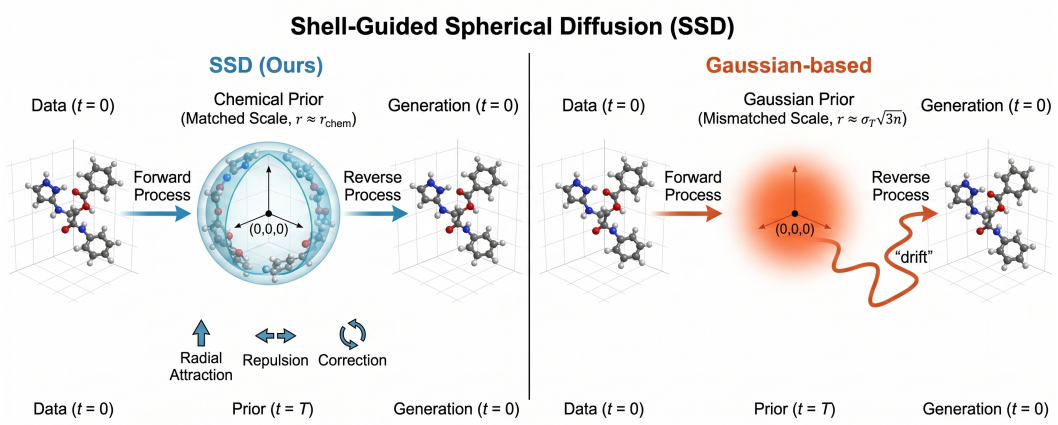


Figure 1: **SSD vs. Gaussian-based diffusion.** In both panels, r denotes the initialization radius: $r = r_{\text{SSD}} \approx r_{\text{chem}}$ for SSD (chemically calibrated), and $r = r_{\text{Gauss}} \approx \sigma_T\sqrt{3n}$ for Gaussian baselines (set by the noise level and atom count, where n is the number of atoms and σ_T is the terminal noise level). (Left) SSD initializes on a chemically scaled spherical shell and follows a direct, stable pathway. The reverse process is guided by three structured drifts—radial attraction, short-range repulsion, and SE(3)-equivariant correction—shown below the shell. (Right) Gaussian-based models start from a prior whose per-atom scale is set by the noise schedule rather than by chemistry; the resulting absolute transport discrepancy grows with $\sqrt{n}$, producing long, wandering trajectories.

and yields long, scale-inconsistent denoising trajectories, with an absolute transport discrepancy that grows with $\sqrt{n}$ (Section 4.5).

The annulus phenomenon itself is universal; it matters here because two properties hold jointly. First, the data scale is intrinsic: it is set by chemistry, whereas σ_T is fixed by the diffusion formulation, and rescaling σ_T to match it would change the signal-to-noise ratio at every t and hence the schedule the backbone was designed around. Second, the ambient dimension $3n$ varies across instances, so the absolute correction the reverse process must make grows with molecule size. Fixed-resolution images satisfy neither property. Domains that satisfy both—3D point clouds and shapes, protein and peptide backbones, crystals and materials, and set-structured data generally—are natural candidates for a scale-calibrated prior.

To address this issue, we propose **Shell-guided Spherical Diffusion (SSD)**, a geometry-aware framework that introduces structure into both initialization and dynamics (Figure 1, Left). In contrast to Gaussian baselines that combine an isotropic prior with unstructured, undirected noise, SSD pairs a chemically scaled spherical-shell prior with structured, directional, geometry-aware dynamics. Specifically, SSD replaces the Gaussian prior with a dataset-adaptive spherical shell that matches the typical radius of training conformations. Shell initialization alone is insufficient; SSD therefore couples it with *shell-aware dynamics*: (i) forward and reverse *radial attraction* that aligns global attraction with molecular scale, (ii) *short-range repulsion* that enforces a minimum separation $d_{\min}$, and (iii) an *SE(3)-equivariant correction* that refines local structure. Together, these components suppress long-range Gaussian drift, reduce spatial entropy, and produce a consistent shell-to-structure generative pathway. We further extend these ideas to deterministic ODE generators via **SSD-Flow**, a coordinate-space flow matching model that incorporates the same shell prior and geometry-aware drifts.

SSD provides a unified geometry-aware formulation applicable across coordinate-based molecular generators (see Appendix N for detailed scope discussion). Our main contributions are: **(i) Shell-aligned formulation:** We identify the Gaussian concentration radius as a non-chemical artifact and introduce a chemically scaled shell with a shell-directed training path and a geometry-guided hybrid reverse sampler, whose radial stability we verify empirically (Sec. 3, Appendix J.3). **(ii) Paired backbone improvements:** SSD improves five representative backbones—GeoDiff [Xu et al., 2022], SubGDiff [Zhang et al., 2024], EDM [Hoogeboom et al., 2022], SemlaFlow-style flow matching [Irwin et al., 2025], and MCF [Wang et al., 2024]—in paired comparisons under identical training and sampling budgets; the large gains are stable across three training seeds, while the smallest

(MCF on GEOM-Drugs) is not statistically significant (Sec. 4.1, Table 4). **(iii) Efficiency, stability, and scalability:** SSD accelerates convergence (Sec. 4.3), remains robust under reduced step budgets (Sec. 4.4), reduces spatial entropy (Sec. 4.6), and yields *growing* gains on larger molecules (e.g., $+32.5$ COV-R for ≥ 50-atom GEOM-Drugs molecules; Sec. 4.5, Appendix L.6), where Gaussian baselines degrade most. To facilitate reproducibility, we release all code, datasets, and pretrained checkpoints, available at[1].

2 Preliminaries

We represent a molecule by its atomic coordinates $\mathbf{x}_0 = \{\mathbf{x}_0^{(i)}\}_{i=1}^n \in \mathbb{R}^{n\times 3}$, where n is the number of atoms. Molecular conformations are defined only up to global translations and rotations, motivating the use of isotropic noise processes and SE(3)-equivariant neural architectures in coordinate-space generative models.

2.1 Task settings and geometric structure

Coordinate-based diffusion models are used in two settings: (i) *conditional generation*, which maps a molecular graph to a 3D conformation; and (ii) *unconditional refinement*, which denoises a noisy geometry into a chemically valid structure. Both settings use a Gaussian prior $\mathbf{x}_T \sim \mathcal{N}(0, \sigma_T^2 I)$, whose Frobenius-norm concentration radius $\sigma_T\sqrt{3n}$ (Appendix B) is mismatched to molecular scale, motivating SSD's spherical-shell initialization.

2.2 SE(3)-equivariance

Since molecular structures have no absolute position or orientation, a coordinate-space generative model should transform consistently under the special Euclidean group. For a rigid motion $g(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$ with $R \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$, a function $f : \mathbb{R}^{n\times 3} \to \mathbb{R}^{n\times 3}$ is *SE(3)-equivariant* if $f(g(\mathbf{x})) = g(f(\mathbf{x}))$. SE(3)-equivariance is satisfied by existing backbones such as GeoDiff [Xu et al., 2022], SubGDiff [Zhang et al., 2024], and EDM [Hoogeboom et al., 2022]. SSD preserves this symmetry because all additional drift terms depend only on normalized directions or pairwise displacements, which rotate covariantly and are translation-invariant.

2.3 Diffusion models

Denoising diffusion probabilistic models [Ho et al., 2020] define a forward SDE $d\mathbf{x}_t = f(\mathbf{x}_t, t)\,dt + g(t)\,d\mathbf{w}_t$ and a corresponding reverse-time SDE [Song et al., 2021]: $d\mathbf{x}_t = \big(f(\mathbf{x}_t, t) - g(t)^2\, s_\theta(\mathbf{x}_t, t)\big)\,dt + g(t)\,d\bar{\mathbf{w}}_t$, where s_θ approximates $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. Removing the stochastic term yields the deterministic probability-flow ODE.

Relation to SSD. SSD follows the same diffusion formalism but replaces the Gaussian prior with a chemically scaled spherical-shell initialization, trains on a shell-directed corruption path (*forward radial attraction*), and samples with a geometry-guided hybrid reverse process that adds *reverse radial attraction* and *short-range repulsion* to the learned *SE(3)-equivariant correction*. The reverse process is not derived as the exact time reversal of the corruption path; its radial stability is verified empirically (Appendix J.3).

3 Methodology

Shell-guided Spherical Diffusion (SSD) consists of two jointly designed components: (i) a *spherical-shell initialization* that replaces the Gaussian prior, and (ii) *shell-aware drift fields* in both forward and reverse processes. Together, they constrain the generative trajectory to a stable, low-entropy, shell-to-structure pathway.

Centering and shell initialization. To remove global translations, every state is centered to zero center of mass: $\mathbf{x} \leftarrow \mathbf{x} - \frac{1}{n}\sum_{i=1}^n \mathbf{x}^{(i)}$. SSD then initializes all atoms on a spherical shell of dataset-adaptive radius r_{SSD}, calibrated to the mean radius of centered training conformations

[1]Code and datasets for SSD are available at: `https://github.com/ShellDiffusion/SSD-ShellDiffusion`.

(Figure 1, Table 9); we write r for r_{SSD} in this section. Crucially, r_{SSD} is a *fixed dataset-level constant*: a per-molecule radius would make p_T conditional on molecular identity, breaking its role as a valid unconditional marginal in the diffusion formalism, and empirically yields $< 0.3\%$ improvement (Appendix L.5). We draw n random Gaussian vectors and normalize each to radius r: $\mathbf{x}_T^{(i)} = r\,\mathbf{v}^{(i)}/\|\mathbf{v}^{(i)}\|$, $\mathbf{v}^{(i)} \sim \mathcal{N}(0, I)$, and then re-center the resulting configuration. After centering, the empirical center of mass is zero by construction, while individual atomic radii deviate from r by $O(r/\sqrt{n})$—a residual that is absorbed into the thin-shell thickness τ analyzed in Appendix H. The measured relative thickness is $\tau/r \leq 0.25$ on QM9 and Drugs (Appendix H). Atoms are assigned via a random permutation π resampled per trajectory; this is essential—deterministic nearest-distance matching collapses performance (Appendix L.4).

Forward SDE. The forward process replaces Gaussian drift with a normalized directional attraction that guides each atom toward its assigned shell point at uniform speed α_t:

$$f^{(i)}(\mathbf{x}_t, t) = \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|}. \tag{1}$$

The normalization yields a *uniform absolute drift speed*: each atom's distance to its assigned shell point contracts linearly at rate α_t, independent of the initial distance. This contrasts with Ornstein–Uhlenbeck dynamics, whose exponential restoring force makes the convergence rate scale with the current distance, producing a wide spread of effective timescales when atoms are initialized at different distances. The linear, distance-independent contraction yields a tighter shell concentration during the late forward phase (formal analysis in Appendix J). The forward SDE is $d\mathbf{x}_t^{(i)} = f^{(i)}(\mathbf{x}_t, t)\,dt + \sigma_t\,d\mathbf{w}_t^{(i)}$ with $\sigma_t \ll 1$. With $\sigma_t = 0$, all atoms reach the radius-r shell exactly; with small $\sigma_t > 0$ and after centering, the resulting forward marginal is a thin spherical layer of thickness $\tau \ll r$ jointly determined by Brownian broadening and the $O(r/\sqrt{n})$ centering residual (Appendix H). During training, this SDE is not simulated: samples are drawn in closed form around its deterministic ($\sigma_t = 0$) path, $\mathbf{x}_t = \bar{\mathbf{x}}_t + \tau_t \boldsymbol{\epsilon}$ (Appendix J.5, item C5).

Reverse SDE. SSD initializes the reverse process from the same shell distribution, ensuring a direct geometric path from shell to structure. The reverse process is a *geometry-guided hybrid sampler*: it is not derived as the time reversal of the forward process. The reverse drift consists of three components: $d\mathbf{x}_t^{(i)} = \left[\mathbf{v}_{\text{rad}}^{(i)} + \mathbf{v}_{\text{rep}}^{(i)} + \mathbf{v}_{\text{score}}^{(i)}\right] dt + \sigma_t\,d\bar{\mathbf{w}}_t^{(i)}$, where (i) *radial attraction* guides the shell inward at uniform speed $\mathbf{v}_{\text{rad}}^{(i)} = -\alpha_t\,\mathbf{x}_t^{(i)}/\|\mathbf{x}_t^{(i)}\|$, mirroring the forward dynamics; (ii) *score-based correction* refines local geometry via an SE(3)-equivariant network $\mathbf{v}_{\text{score}}^{(i)} = \sigma_t^2\,s_\theta(\mathbf{x}_t, t)^{(i)}$; and (iii) *short-range repulsion* prevents atomic collapse by enforcing minimum separation $d_{\min}$:

$$\mathbf{v}_{\text{rep}}^{(i)} = \alpha_t \sum_{j \neq i} \mathbb{I}(d_{ij} < d_{\min})\,\frac{d_{\min} - d_{ij}}{d_{\min}}\,\frac{\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}}{d_{ij}}, \quad \text{where } d_{ij} = \|\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}\|. \tag{2}$$

Unified radial framework. Both forward and reverse processes employ normalized *radial attraction*—drift along the direction connecting atoms to target points. Forward: each atom targets its assigned shell point $\mathbf{x}_T^{(\pi(i))}$ (distributed radial directions); Reverse: all atoms target the shell center (unified inward contraction). This symmetric design replaces the exponential convergence heterogeneity of OU dynamics with linear, distance-independent contraction, yielding tightly concentrated, geometry-aligned trajectories (deterministic analysis in Appendix J.1).

Training, inference, and equivariance. SSD instantiates two training paradigms depending on the backbone: *denoising score matching* for diffusion-based backbones (SSD-GeoDiff, SSD-SubGDiff, SSD-EDM, SSD-MCF), and *flow matching* for SSD-Flow. In both cases, the corruption process (or probability path) is built from the SSD forward dynamics rather than a Gaussian, ensuring the learned network is geometry-aware from training rather than only at inference. The training loss *form* and optimization remain unchanged from each backbone; only the data corruption mechanism and the corresponding regression target differ (details in the Appendix K); SSD-Flow additionally uses a coordinate-space encoder (Appendix M.1). Inference uses the same shell initialization, with the stochastic term optionally kept for thin-shell diversity or removed for deterministic sampling; SSD remains stable under reshuffled atom–shell assignments, confirming it does not rely on brittle ordering

heuristics. Finally, SSD preserves SE(3)-equivariance by construction: all added drift terms depend only on normalized directions or pairwise displacements, which transform covariantly under rotations and are translation-invariant (proof in Appendix I). A detailed clarification of the relationship between SSD's forward, reverse, and training procedures and standard score-based diffusion theory—including the role of the OU description, the choice of isotropic reverse drift, training under SSD forward marginals, and the role of $\mathbf{v}_{\mathrm{rep}}$—is provided in Appendix J.5.

4 Experiments

Evaluation Settings and Datasets. We evaluate SSD on the GEOM-QM9 [Ramakrishnan et al., 2014] and GEOM-Drugs [Axelrod and Gomez-Bombarelli, 2022] benchmarks, the standard testbeds for coordinate-space molecular generation. The two span markedly different geometric scales—$r_{\mathrm{SSD}} = 6.37$ Å for QM9 (small rigid molecules) vs. 21.16 Å for Drugs (large flexible drug-like structures), a $3.3\times$ range; SSD is calibrated and trained separately on each. Both are adopted by nearly all coordinate-space backbones [Xu et al., 2022, Zhang et al., 2024, Hoogeboom et al., 2022, Hua et al., 2024, Wang et al., 2024, Morehead and Cheng, 2024, Irwin et al., 2025], enabling fair comparison under identical preprocessing (Appendix E; the scope of these results with respect to distribution shift is discussed in Appendix L.8).

Evaluation Metrics. Each backbone and its SSD-enhanced variant is evaluated strictly under its canonical protocol. Conditional models (GeoDiff, SubGDiff, and their SSD counterparts) follow the SubGDiff evaluation protocol [Zhang et al., 2024]; unconditional refinement models (EDM, SemlaFlow, SSD-EDM, SSD-Flow) follow the SemlaFlow protocol [Irwin et al., 2025]; and MCF/SSD-MCF follow the original Conformer Fields protocol [Wang et al., 2024]. Across all settings, thresholds, alignment rules, RMSD criteria, and duplicate filtering remain unchanged. We report COV-R/COV-P (diversity, ↑) and MAT-R/MAT-P in Å (quality, ↓) for conditional generation, and Atom/Mol Stability, Validity, and Uniqueness (all ↑, in %) for unconditional refinement. Unless otherwise specified, COV and MAT scores are reported as means; full definitions are provided in Appendix D.

Computational Cost. SSD introduces no additional learnable parameters for coordinate-space diffusion backbones, and its added drift fields (radial attraction and repulsion) are closed-form and computationally negligible. Training and inference time therefore remain effectively unchanged (e.g., SubGDiff [Zhang et al., 2024] → SSD-SubGDiff). For latent-flow architectures, SSD-Flow replaces SemlaFlow's [Irwin et al., 2025] heavier encoder with a lightweight EDM-style [Hoogeboom et al., 2022] encoder while keeping the flow formulation, network depth, and number of function evaluations (NFEs) identical. This reduces computational cost without altering the underlying refinement dynamics. A full parameter and wall-clock comparison is provided in Appendix M.

Baselines and Training Settings. We evaluate SSD on five representative coordinate-space backbones: GeoDiff [Xu et al., 2022], SubGDiff [Zhang et al., 2024], EDM [Hoogeboom et al., 2022], SemlaFlow-style flow matching [Irwin et al., 2025], and MCF [Wang et al., 2024]. We follow their official implementations and protocols exactly to isolate SSD's contribution. SSD acts as a plug-in enhancement that replaces Gaussian initialization with spherical-shell priors, replaces the Gaussian training corruption with the SSD corruption path and its regression target, and adds structured reverse dynamics; for the score-based backbones it keeps the denoiser architecture, loss form, and sampling budget. We additionally introduce **SSD-Flow**, which adapts SemlaFlow to coordinate-space flow matching on the SSD shell path, replacing SemlaFlow's encoder with an EDM-style coordinate encoder while retaining the original substructure features. Detailed baseline descriptions, coordinate updates, and training configurations are provided in Appendices C, F, and K.

4.1 Main Results

This section summarizes our main empirical findings. SSD provides consistent improvements across five representative coordinate-space backbones—GeoDiff [Xu et al., 2022], SubGDiff [Zhang et al., 2024], EDM [Hoogeboom et al., 2022], SemlaFlow-style flow matching [Irwin et al., 2025], and MCF [Wang et al., 2024]—each evaluated under the standard protocol used for its category (SubGDiff protocol for GeoDiff/SubGDiff, SemlaFlow protocol for EDM/SemlaFlow, and the original MCF protocol for MCF). All models are trained with identical preprocessing, hyperparameters, and

Table 1: Results for conditional generation under SubGDiff protocol (top), conditional generation under MCF protocol (middle), and unconditional refinement under SemlaFlow protocol (bottom). GeoDiff and SubGDiff are evaluated under the SubGDiff protocol; EDM and SemlaFlow follow the SemlaFlow protocol. MCF is evaluated under its original MCF protocol, as the original work did not report results under the SubGDiff protocol. Each backbone is shown in both its original form and its SSD-augmented variant. Bold indicates improvement over the corresponding backbone, and "–" denotes metrics not reported in the original papers. SSD entries are single runs (training seed 42); three-seed statistics for representative pairs are in Table 4.

Conditional Generation (QM9 & Drugs) — SubGDiff Protocol									
Model	Dataset	COV-R ↑		COV-P ↑		MAT-R ↓		MAT-P ↓	
		mean	median	mean	median	mean	median	mean	median
GeoDiff	QM9	80.36	83.82	53.66	50.85	0.2820	0.2799	0.6673	0.4214
SSD-GeoDiff	QM9	**92.00**	**93.80**	**61.40**	**59.75**	**0.2425**	**0.2508**	**0.4188**	**0.4136**
GeoDiff	Drugs	64.12	75.56	52.79	50.29	–	–	–	–
SSD-GeoDiff	Drugs	**91.69**	**94.86**	**70.23**	**68.11**	0.8463	0.8247	1.1543	1.0894
SubGDiff	QM9	90.91	95.59	50.16	48.01	0.2460	0.2351	0.6114	0.4791
SSD-SubGDiff	QM9	**93.20**	**96.10**	**63.50**	**61.80**	**0.2380**	**0.2300**	**0.4100**	**0.4050**
SubGDiff	Drugs	76.16	86.43	–	–	1.0003	0.9905	–	–
SSD-SubGDiff	Drugs	**92.50**	**95.80**	71.10	69.00	**0.8300**	**0.8100**	1.1400	1.0700
Conditional Generation (QM9 & Drugs) — MCF Protocol									
MCF	QM9	95.00	100.00	93.70	100.00	0.1030	0.0440	0.1190	0.0550
SSD-MCF	QM9	**96.15**	100.00	**94.61**	100.00	**0.0993**	**0.0437**	**0.1087**	**0.0536**
MCF	Drugs	84.70	92.20	66.80	71.30	0.3900	0.2470	0.6180	0.5300
SSD-MCF	Drugs	**85.37**	**92.81**	**68.82**	**74.26**	**0.3792**	**0.2401**	**0.6089**	**0.5263**
Unconditional Refinement (QM9 & Drugs) — SemlaFlow Protocol									
Model	Dataset	AtomStab ↑	MolStab ↑	Valid ↑	Unique ↑				
EDM	QM9	98.7	82.0	91.9	98.9				
SSD-EDM	QM9	**99.3**	**99.8**	**100.0**	**100.0**				
EDM	Drugs	81.3	–	–	–				
SSD-EDM	Drugs	**90.5**	70.1	98.5	99.2				
SemlaFlow	QM9	99.9	99.7	99.4	95.4				
SSD-Flow	QM9	99.9	**99.9**	**100.0**	**100.0**				
SemlaFlow	Drugs	99.8	97.3	93.9	100.0				
SSD-Flow	Drugs	99.8	**97.6**	**100.0**	100.0				

Table 2: Ablation study on GEOM-QM9 (SubGDiff protocol). We systematically ablate initialization (Gaussian vs. spherical shell) and dynamics (baseline vs. SSD components). $r_{\text{Gauss}} = \sigma_T\sqrt{3n_{\text{med}}}$ denotes the Gaussian concentration shell; r_{SSD} denotes SSD's data-adaptive radius. Full SSD dynamics correspond to $\mathbf{v}_{\text{rad}} + \mathbf{v}_{\text{rep}} + \mathbf{v}_{\text{score}}$. $\mathbf{v}_{\text{SubGD}}$ denotes SubGDiff's original Gaussian-based reverse drift. Single run (training seed 42); the same protocol on GEOM-Drugs and on SSD-EDM and SSD-MCF is in Table 3.

Variant	**Init./Prior**	**Dynamics**	**COV-R ↑**	**COV-P ↑**	**MAT-R ↓**	**MAT-P ↓**
(1) Gaussian baseline	Gaussian	$\mathbf{v}_{\text{SubGD}}$	90.91	50.16	0.2460	0.6114
(2) Gaussian + rep./corr.	Gaussian	$\mathbf{v}_{\text{rep}} + \mathbf{v}_{\text{score}}$	90.95	50.20	0.2455	0.6098
(3) Shell(r_{Gauss})	Shell(r_{Gauss})	None	90.92	50.18	0.2462	0.6107
(4) Shell(r_{Gauss}) + SSD dyn.	Shell(r_{Gauss})	$\mathbf{v}_{\text{rad}} + \mathbf{v}_{\text{rep}} + \mathbf{v}_{\text{score}}$	91.50	51.00	0.2443	0.6020
(5) SSD (ours)	**Shell(r_{SSD})**	$\mathbf{v}_{\text{rad}} + \mathbf{v}_{\text{rep}} + \mathbf{v}_{\text{score}}$	**93.20**	**63.50**	**0.2380**	**0.4100**

sampling budgets. As shown in Table 1, SSD strengthens every backbone without adding parameters or increasing training cost. Three-seed statistics (Table 4) confirm the SubGDiff, EDM, and SSD-Flow gains; the MCF gain on GEOM-Drugs ($+0.76$) is not statistically significant. In particular, weaker coordinate-based models such as GeoDiff and EDM benefit substantially from shell-guided dynamics and achieve performance comparable to, or surpassing, larger diffusion, VAE, and flow-based methods as reported in their papers; these cross-paper comparisons are secondary to the paired ones.

To demonstrate model-agnostic behavior, Appendix G reports extended comparisons with a broader range of molecular generative models, including diffusion-, equivariant-, and flow-based approaches [Xu et al., 2021a,b, Ganea et al., 2021, Jing et al., 2022, Wang et al., 2024, Morehead and Cheng, 2024, Dunn and Koes, 2024, Xu et al., 2023, Hua et al., 2024]. Across these diverse architectures and training regimes, the same pattern emerges: SSD consistently improves coordinate-space backbones when evaluated under their native protocols. A controlled Gaussian-Flow baseline (Appendix M.1) isolates

Table 3: The five-variant ablation of Table 2 repeated on GEOM-Drugs (SSD-SubGDiff) and on two further backbones on GEOM-QM9 (SSD-EDM, SSD-MCF). Variant definitions are unchanged; rows (1) and (5) are the Table 1 endpoints. Single runs.

Variant	**Drugs COV-R ↑** (SSD-SubGDiff)	**QM9 MolStab ↑** (SSD-EDM)	**QM9 COV-R ↑** (SSD-MCF)
(1) Gaussian baseline	76.2	82.0	95.00
(2) Gaussian + rep./corr.	76.4	82.4	95.05
(3) Shell(r_{Gauss})	76.3	82.1	95.02
(4) Shell(r_{Gauss}) + SSD dyn.	79.1	86.0	95.40
(5) SSD (r_{SSD})	**92.5**	**99.8**	**96.15**

Table 4: Three-seed means ± standard deviations (training seeds 42/43/44; standard deviation across runs) for representative paired comparisons. Seed 42 is the run reported in Table 1. For EDM, the QM9 column reports MolStab and the Drugs column AtomStab. The MCF gain on GEOM-Drugs ($+0.76$, $\approx 1.8\sigma$ of the paired difference) is sign-consistent but not statistically significant.

Pair	**Metric**	**QM9: base → SSD**	**Drugs: base → SSD**
SubGDiff / SSD-SubGDiff	COV-R ↑	$90.94 \pm 0.21 \rightarrow 93.16 \pm 0.18$	$76.28 \pm 0.44 \rightarrow 92.44 \pm 0.33$
EDM / SSD-EDM	MolStab / AtomStab ↑	$82.3 \pm 0.6 \rightarrow 99.7 \pm 0.1$	$81.0 \pm 0.5 \rightarrow 90.6 \pm 0.4$
MCF / SSD-MCF	COV-R ↑	$95.04 \pm 0.15 \rightarrow 96.11 \pm 0.12$	$84.66 \pm 0.31 \rightarrow 85.42 \pm 0.27$
Gaussian-Flow / SSD-Flow	Valid ↑	$97.65 \pm 0.22 \rightarrow 99.95 \pm 0.05$	–
Gaussian-Flow / SSD-Flow	Unique ↑	$95.14 \pm 0.26 \rightarrow 99.96 \pm 0.04$	–

the shell prior from the encoder choice, attributing $+2.4/+4.9/+1.8$ on Validity/Uniqueness/MolStab to the shell prior; the encoder swap alone lowers performance, so the coordinate encoder and the shell prior are jointly required.

4.2 Ablation Study

A core requirement of SSD is that the forward and reverse processes share a geometrically consistent radius. A raw Gaussian prior makes this difficult to test directly: under $\mathcal{N}(0, \sigma_T^2 I_{3n})$, samples concentrate around radius $\sigma_T\sqrt{3n}$, which varies per molecule through n. Applying SSD's reverse radial attraction $\mathbf{v}_{\text{rad}} = -\alpha_t \mathbf{x}_t / \|\mathbf{x}_t\|$ to such per-molecule-varying samples would create asymmetric drift across molecular sizes and confound the comparison. We therefore construct a controlled *Gaussian-shell* variant by projecting Gaussian samples onto a fixed reference shell at $r_{\text{Gauss}} = \sigma_T\sqrt{3n_{\text{med}}}$, using the median atom count n_{med} to define a single dataset-wide radius derived from dimensionality rather than chemistry. This makes SSD's radial attraction $\mathbf{v}_{\text{rad}}$ applicable to both Gaussian-derived and chemically calibrated shells, separating the effect of radius *scale* from that of shell-based initialization itself (Figure 1).

Table 2 reports five controlled variants: (1) The Gaussian baseline corresponds to the original SubGDiff model, where each molecule's samples naturally concentrate around its dimension-dependent radius $\sigma_T\sqrt{3n}$. (2) Adding repulsion and equivariant correction on top of the unconstrained Gaussian prior yields minimal change, confirming that guidance terms alone cannot correct the radius mismatch. (3) Projecting the Gaussian prior onto a fixed shell at $r_{\text{Gauss}} = \sigma_T\sqrt{3n_{\text{med}}}$ produces nearly identical performance to (1), demonstrating that unconstrained Gaussians already behave as dimension-dependent shells. (4) Applying full SSD dynamics on the Gaussian shell yields modest gains but remains far below SSD, showing that even with shell-consistent dynamics, the dimensional scaling r_{Gauss} is fundamentally misaligned with molecular scale. (5) Full SSD—combining a chemically calibrated shell at radius r_{SSD} with shell-consistent dynamics—achieves the strongest performance. Two further principles confirmed in Appendix L round out the picture: shell geometry *cannot* be replicated by noise tuning—reducing Gaussian variance consistently degrades performance—and random atom–shell assignments are essential, as deterministic nearest-matching introduces brittle ordering dependencies that collapse coverage at inference. Only SSD aligns initialization, noise structure, and reverse dynamics into a single geometry-consistent framework. Extended ablations covering radius calibration, stratified evaluation across molecular scales, and the scope of these results with respect to distribution shift are provided in Appendices L.5–L.8.

Table 3 repeats the five-variant protocol on GEOM-Drugs (SSD-SubGDiff) and on two further backbones (SSD-EDM and SSD-MCF on QM9). The ordering of the variants is identical in all three settings: guidance terms on a Gaussian prior (2) and a dimension-derived shell without dynamics (3) change little, shell-aware dynamics on the dimension-derived shell (4) recover a minority of the gain, and the calibrated radius (5 vs. 4, dynamics held fixed) accounts for the largest step—+1.7 COV-R on QM9 and +13.4 on Drugs for SSD-SubGDiff. The effect persists on MCF, which starts at 95.00 COV-R.

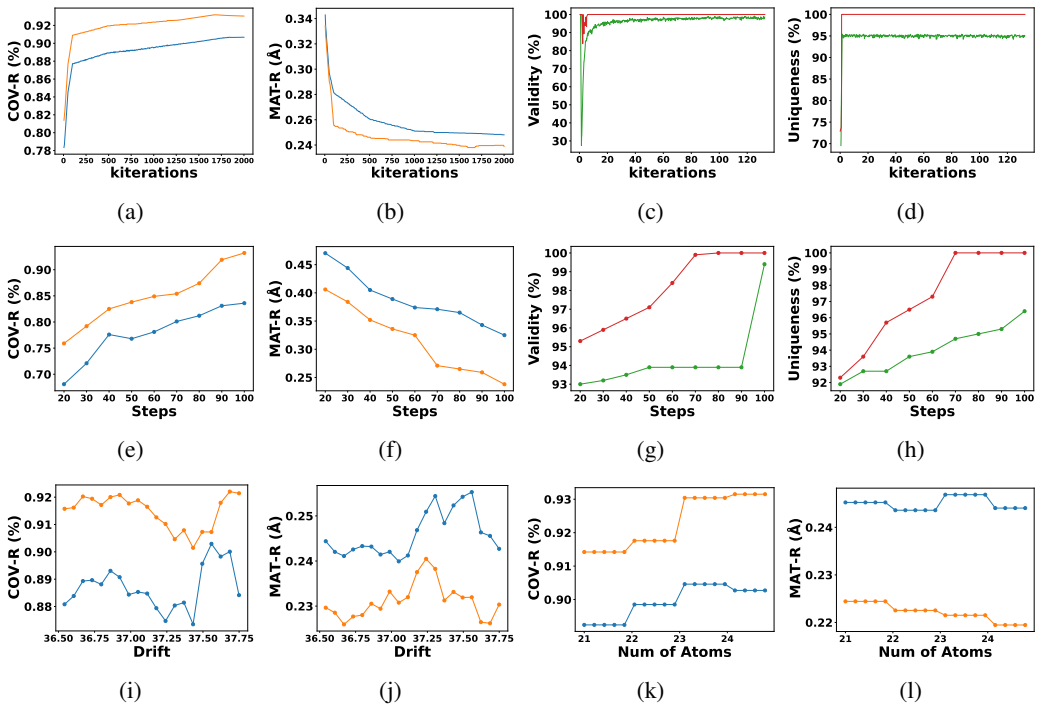


Figure 2: **SSD on QM9: convergence, step robustness, and long-range drift.** (Color coding: Orange = SSD-SubGDiff, Blue = SubGDiff, Red = SSD-Flow, Green = SemlaFlow.) **Row 1 — Faster convergence (a–d).** (a,b) Conditional generation (SSD-SubGDiff vs. SubGDiff), COV-R (↑) and MAT-R (↓) over training iterations. (c,d) Unconditional refinement (SSD-Flow vs. SemlaFlow), Validity (↑) and Uniqueness (↑) over training iterations. SSD converges faster and consistently outperforms Gaussian-based baselines across both tasks. **Row 2 — Robust improvements across sampling steps (e–h).** "Steps" denotes the number of sampling steps, unified across diffusion (steps or ODE evaluations) and flow matching (NFEs). (e,f) Conditional generation, COV-R (↑) and MAT-R (↓) under varied step budgets. (g,h) Unconditional refinement, Validity (↑) and Uniqueness (↑) under varied step budgets. **Row 3 — Case study on long-range Gaussian drift (i–l).** SSD-SubGDiff begins to consistently outperform SubGDiff once drift exceeds ∼36.6 Å (i,j) or the molecule contains more than 21 atoms (k,l), achieving higher COV-R (↑) and lower MAT-R (↓). Training curves in rows 1–2 are reproduced using the official implementations of SubGDiff [Zhang et al., 2024] and SemlaFlow [Irwin et al., 2025].

4.3 Faster Convergence with SSD

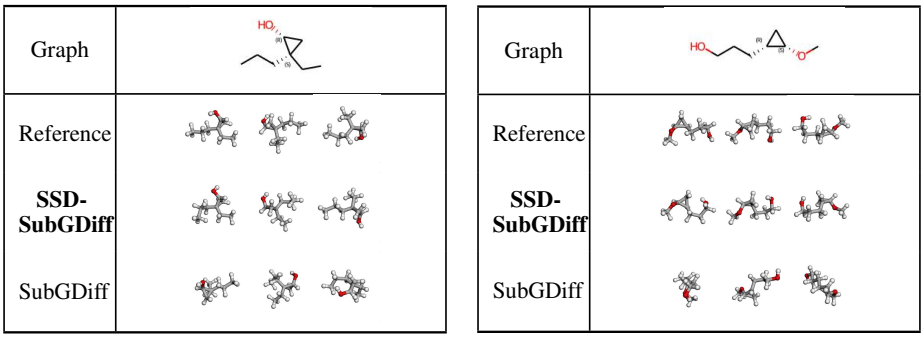
A key advantage of SSD is faster convergence than Gaussian-based baselines. Figures 2a–2d show comparisons on QM9 for both conditional generation and unconditional refinement. For conditional generation, SSD-SubGDiff achieves higher COV-R and lower MAT-R across training iterations compared to SubGDiff [Zhang et al., 2024], simultaneously improving *diversity* (higher COV-R) and *quality* (lower MAT-R). For unconditional refinement, SSD-Flow reaches near-perfect validity and uniqueness much earlier than SemlaFlow. Compact shell initialization produces clearer denoising gradients than highly dispersed Gaussian noise, accelerating training across both score-matching and flow-matching paradigms.

4.4 Robustness Across Step Budgets

Beyond faster training convergence, SSD also improves sampling efficiency by being more robust to step budgets. Figures 2e–2h report QM9 results across 20–100 sampling steps—diffusion timesteps for score-matching models and ODE function evaluations for flow-matching models [Irwin et al., 2025]. SSD-SubGDiff outperforms SubGDiff [Zhang et al., 2024] on COV-R and MAT-R across all settings, and SSD-Flow achieves substantially higher validity and uniqueness than SemlaFlow even with very few steps. SSD therefore preserves both *diversity* and *quality* under aggressive step reduction, an important property for efficiency-critical applications such as large-scale virtual screening.

4.5 Case Study: Mitigating Long-Range Gaussian Drift

Gaussian priors are particularly problematic for larger and more flexible molecules: although the per-atom mismatch between the prior and the data need not grow with n, the total displacement the reverse process must correct grows as $\sqrt{n}$, destabilizing refinement. We examine COV-R and MAT-R as functions of total drift (displacement from initialization to the final conformation) and the number of atoms. Figures 2i–2l show that SSD-SubGDiff consistently outperforms SubGDiff once drift exceeds $\sim$36.6 Å or molecules contain more than 21 atoms; in these regimes SSD maintains higher *quality* (lower MAT-R) and greater *diversity* (higher COV-R), while SubGDiff rapidly deteriorates. Figure 3 confirms qualitatively that SSD produces stable, chemically plausible structures under both long-drift and large-molecule settings, whereas SubGDiff often yields distorted intermediates. The joint effect of shell initialization and structured reverse dynamics directly addresses the failure mode of Gaussian-based methods at scale.



(a) Long-drift regime.

(b) Molecules with >21 atoms.

Figure 3: **Qualitative case studies on QM9 (conditional generation). (a)** Long-drift regimes (drift > 36.6 Å); **(b)** Molecules with >21 atoms. SSD-SubGDiff produces stable, chemically plausible structures matching the reference, while SubGDiff often yields distorted or unstable geometries.

4.6 Trajectory Stability and Spatial Entropy

Beyond final accuracy, an effective generative model should produce stable, geometry-aligned sampling trajectories. Gaussian-based diffusion often wanders due to the mismatch between the prior radius and the true molecular scale. To measure this effect across both SDE-based (e.g., SubGDiff) and ODE-based (e.g., SemlaFlow / SSD-Flow) samplers, we use two geometry-aware metrics: **(i) Radial deviation** $\mathbb{E}_t[\,|\|x_t\| - r_{\text{SSD}}|\,]$, which quantifies how far a trajectory strays from the chemically meaningful shell radius r_{SSD}; and **(ii) Path-excess ratio** $\text{PathLen}/\|x_T - x_0\|$, comparing the actual trajectory length to the direct shell-to-structure displacement. As shown in Table 5, SSD reduces both quantities substantially. The strict like-for-like comparison is the ODE-vs-ODE row (SemlaFlow vs. SSD-Flow), where both samplers are deterministic and the residual gap therefore isolates the geometric contribution of the shell-aligned path; the larger SDE-vs-SDE gap on path excess additionally reflects SSD's smaller thin-shell noise. Radial deviation, which is independent of injected noise scale, drops by a similar relative factor in both regimes (SDE: $0.45 \to 0.18$; ODE: $0.21 \to 0.11$), confirming that the gain is geometric rather than a stochasticity artifact. Together,

these metrics show that SSD produces low-entropy trajectories aligned with chemically meaningful radii, complementing the performance gains in Sec. 4.1.

Table 5: Trajectory-stability metrics on GEOM-QM9. SSD yields lower-entropy, geometry-aligned trajectories in both SDE (SubGDiff) and ODE (SemlaFlow) regimes.

Method	**Regime**	Radial dev. ↓	Path excess ↓
SubGDiff	SDE	0.4534	246.91
SSD-SubGDiff	SDE	**0.1781**	**1.71**
SemlaFlow	ODE	0.2145	1.29
SSD-Flow	ODE	**0.1096**	**1.11**

5 Related Work

Prior approaches to 3D molecular generation fall into three broad families. (i) Coordinate-space diffusion models such as GeoDiff [Xu et al., 2022], EDM [Hoogeboom et al., 2022], SubGDiff [Zhang et al., 2024], MuDiff [Hua et al., 2024], and GCDM [Morehead and Cheng, 2024] diffuse atomic coordinates with SE(3)-equivariant networks, but all adopt Gaussian priors, leading to dispersed initial states and long denoising trajectories. (ii) Latent- and flow-based models such as GeoLDM [Xu et al., 2023], SemlaFlow [Irwin et al., 2025], FlowMol [Dunn and Koes, 2024], and FlowMol3 [Dunn and Koes, 2025] improve efficiency through flow matching, yet still rely on Gaussian priors. SSD provides a plug-in geometric enhancement, replacing Gaussian initialization with chemically scaled shells across coordinate-based backbones, including flow-matching variants like SSD-Flow. (iii) Other distance- or energy-based approaches (e.g., ConfGF [Shi et al., 2021], CGCF [Xu et al., 2021a], GeoMol [Ganea et al., 2021]) emphasize geometric constraints or torsional predictions, which are orthogonal to our focus. A more detailed review of these methods is provided in Appendix A. A complementary line of flow-matching work introduces *harmonic priors* that encode local bonded geometry [Hassan et al., 2024, Jing et al., 2023]; SSD's global shell prior and harmonic priors address different geometric scales and are naturally combinable (Appendix A.4).

6 Conclusion

We proposed **Shell-guided Spherical Diffusion (SSD)**, a model-agnostic, geometry-aware framework that replaces the Gaussian prior with a chemically scaled spherical shell, trains on a shell-directed corruption path, and, for the score-based backbones, samples with a geometry-guided hybrid reverse process, while preserving SE(3)-equivariance and adding no learnable parameters. SSD instantiates two training paradigms (denoising score matching and flow matching); for the score-based backbones it keeps the denoiser architecture and loss form, and SSD-Flow additionally uses a coordinate-space encoder. SSD yields gains that grow with molecule size—reaching $+32.5$ COV-R for ≥ 50-atom molecules. On QM9 and Drugs, SSD improves accuracy, stability, and diversity for all five backbones (GeoDiff, SubGDiff, EDM, SemlaFlow, MCF) in paired comparisons; the large gains are stable across three training seeds, while the MCF gain on GEOM-Drugs is not statistically significant.

Acknowledgments and Disclosure of Funding

We thank Maxora AI, Taiwan, for its support of this work. The authors declare no competing interests.

References

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

Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

Ian Dunn and David R Koes. FlowMol3: Flow matching for 3D de novo small-molecule generation. *arXiv preprint arXiv:2508.12629*, 2025.

Ian Dunn and David Ryan Koes. Mixed continuous and categorical flow matching for 3D de novo molecule generation. *arXiv preprint arXiv:2404.19739*, 2024.

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.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

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. Conformation generation: the state of the art. *Journal of chemical information and modeling*, 57(8):1747–1756, 2017.

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

Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning (ICML)*, 2022.

Chenqing Hua, Sitao Luan, Minkai Xu, Zhitao Ying, Jie Fu, Stefano Ermon, and Doina Precup. MuDiff: Unified diffusion for complete molecule generation. In *Learning on Graphs Conference*. PMLR, 2024.

Ross Irwin, Alessandro Tibo, Jon Paul Janet, and Simon Olsson. Semlaflow–efficient 3d molecular generation with latent attention and equivariant flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 3772–3780. PMLR, 2025.

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.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

Douglas B Kitchen, Hélène Decornez, John R Furr, and Jürgen Bajorath. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nature reviews Drug discovery*, 3 (11):935–949, 2004.

Tuan Le, Julian Cremer, Frank Noé, Djork-Arné Clevert, and Kristof T. Schütt. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. In *International Conference on Learning Representations (ICLR)*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific reports*, 9(1):20381, 2019.

Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation and optimization. *Communications Chemistry*, 7(1):150, 2024.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Z Qiao, F Ding, T Dresselhaus, MA Rosenfeld, X Han, O Howell, A Iyengar, S Opalenski, AS Christensen, and SK Sirumalla. NeuralPLexer3: Accurate biomolecular complex structure prediction with flow models. *arXiv preprint arXiv:2412.10743*, 2024.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

Gisbert Schneider. Automating drug discovery. *Nature reviews drug discovery*, 17(2):97–113, 2018.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Chence Shi, Shitong Luo, Minkai Xu, and Jian Tang. Learning gradient fields for molecular conformation generation. In *International conference on machine learning*, pages 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 *International Conference on Learning Representations (ICLR)*, 2021.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

W Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of chemical research*, 54(2):263–270, 2020.

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

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations (ICLR)*, 2021a.

Minkai Xu, Wujie Wang, Shitong Luo, Chence Shi, Yoshua Bengio, Rafael Gomez-Bombarelli, and Jian Tang. An end-to-end framework for molecular conformation generation via bilevel programming. In *International conference on machine learning*, pages 11537–11547. PMLR, 2021b.

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

Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.

Jiying Zhang, Zijing Liu, Yu Wang, Bin Feng, and Yu Li. Subgdiff: A subgraph diffusion model to improve molecular representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pages 29620–29656, 2024.

A Summary of Prior Work

A.1 Distance- and energy-based methods

ConfGF [Shi et al., 2021], CGCF [Xu et al., 2021a], and ConfVAE [Xu et al., 2021b] generate conformations through distance geometry or energy-inspired objectives. By operating on pairwise distances, these approaches naturally preserve SE(3) invariance. However, they follow a two-stage design (distance prediction $\rightarrow$ coordinate reconstruction), where noise in predicted distances may accumulate during reconstruction, sometimes leading to structural inaccuracies.

A.2 Coordinate-space diffusion models

GeoDiff [Xu et al., 2022], EDM [Hoogeboom et al., 2022], SubGDiff [Zhang et al., 2024], MUDiff [Hua et al., 2024], and GCDM [Morehead and Cheng, 2024] apply diffusion directly to atomic coordinates using SE(3)-equivariant networks. These models commonly employ a Gaussian prior, a simple and widely adopted choice, but one that produces dispersed starting states and long denoising trajectories.

A.3 Latent- and flow-based models

GeoLDM [Xu et al., 2023] introduces a latent diffusion model with invariant and equivariant features to improve efficiency and controllability. SemlaFlow [Irwin et al., 2025] adopts a latent-attention architecture with Flow Matching for 3D molecular generation. FlowMol [Dunn and Koes, 2024] develops a flow-matching framework for mixed continuous and categorical variables. Together, these works highlight the flexibility of latent-space modeling and flow-based generative paradigms, offering complementary design perspectives to coordinate-space diffusion.

A.4 Harmonic priors and complementarity with SSD

A complementary line of work replaces the Gaussian prior with a *harmonic prior*, originally introduced for protein structure prediction in EigenFold [Jing et al., 2023] and adopted in coordinate-space molecular conformer generation by ET-Flow [Hassan et al., 2024] and in biomolecular structure prediction by NeuralPLexer3 [Qiao et al., 2024]. A harmonic prior places atoms according to a Gaussian distribution whose covariance is determined by the graph Laplacian of the molecular bond graph, encoding the inductive bias that bonded atoms should be in close spatial proximity. This is an informative prior that captures *local bonded geometry*—atoms linked by chemical bonds remain near one another at initialization, producing shorter probability paths in flow-matching settings.

Complementarity with SSD. SSD's shell prior and harmonic priors operate at fundamentally different geometric scales and are *complementary*, not competing:

- **SSD shell prior** addresses *global initialization scale*: it corrects the dimensional Gaussian concentration radius $\sigma_T\sqrt{3n}$ by anchoring initialization to the chemically calibrated radius r_{SSD}, ensuring the entire molecule starts at a chemically meaningful spatial extent.
- **Harmonic prior** addresses *local bonded geometry*: it constrains pairs of bonded atoms toward equilibrium proximity, reducing transport cost along bonds during the reverse process.

A natural extension is to combine the two: use SSD's shell to set the correct global scale, then use a harmonic covariance *within* the shell to encode local bonded structure. Because SSD modifies only the initialization distribution and the analytical drift fields, it does not constrain the choice of within-shell covariance, leaving such hybrids fully compatible with the SSD framework.

Why a harmonic-prior baseline is non-trivial in our setting. Harmonic priors require the molecular bond graph to define the Laplacian, which is available in conditional generation but *not* in the unconditional refinement setting evaluated in this paper for EDM/SemlaFlow/SSD-Flow—where the input is a noisy 3D point cloud with no graph structure given. Constructing a harmonic prior in the unconditional setting would require either (i) inferring connectivity from noisy coordinates (introducing an additional bond-prediction step that confounds the comparison), or (ii) restricting

evaluation to the conditional setting where backbones such as ET-Flow already operate. ET-Flow operates in the conditional setting under a different evaluation protocol from SSD-Flow's unconditional refinement setting; bringing the two into a single fair comparison would require unifying these protocols, which we leave as a natural follow-up. A controlled head-to-head ablation that isolates the effect of replacing SSD's shell with a harmonic prior within an otherwise identical pipeline is similarly compatible with the SSD framework and is a natural direction for future work.

B Concentration of the Gaussian Radius

For generality, we use the notation σ_{gauss} to denote the standard deviation of an isotropic Gaussian distribution in $\mathbb{R}^{n\times 3}$. In diffusion-based molecular generation, the prior distribution at terminal time T is typically $p_T(\mathbf{x}) = \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$, so $\sigma_{\text{gauss}} = \sigma_T$. The concentration results below explain why such Gaussian priors produce initialization at Frobenius-norm radius $\sigma_T\sqrt{3n}$. Both this radius and a molecule's own Frobenius norm grow as $\sqrt{n}$; what the Gaussian leaves unmatched is the per-atom scale $\sigma_T\sqrt{3}$, which is set by the noise schedule rather than by molecular geometry.

Let $\mathbf{X} \sim \mathcal{N}(\mathbf{0},\, \sigma_{\text{gauss}}^2 I_{3n})$ and denote $R = \|\mathbf{X}\|_2$. Then $\frac{R}{\sigma_{\text{gauss}}}$ follows a chi distribution with $k = 3n$ degrees of freedom:

$$\frac{R}{\sigma_{\text{gauss}}} \;\sim\; \chi_k, \qquad k = 3n.$$

Classical properties of the chi distribution yield

$$\mathbb{E}[R] = \sigma_{\text{gauss}}\Big(\sqrt{k} \;-\; \tfrac{1}{4\sqrt{k}} + O(k^{-3/2})\Big), \tag{3}$$

$$\mathrm{Var}(R) = \sigma_{\text{gauss}}^2\Big(\tfrac{1}{2} + O(k^{-1})\Big). \tag{4}$$

In particular, the *typical radius*

$$\rho^\star \;:=\; \sigma_{\text{gauss}}\sqrt{k} \;=\; \sigma_{\text{gauss}}\sqrt{3n}$$

approximates $\mathbb{E}[R]$ up to a relative error $O(1/k)$.

Concentration (thin-shell) inequality. The map $f(\mathbf{z}) = \|\mathbf{z}\|_2$ is 1-Lipschitz. By Gaussian concentration,

$$\Pr\Big(\big|R - \mathbb{E}[R]\big| \ge t\Big) \;\le\; 2\exp\Big(-\tfrac{t^2}{2\sigma_{\text{gauss}}^2}\Big).$$

Equivalently, with $t = \varepsilon\,\rho^\star$,

$$\Pr\Big(\big|R - \rho^\star\big| \ge \varepsilon\,\rho^\star\Big) \;\le\; 2\exp\Big(-\tfrac{k\varepsilon^2}{2} + O(\varepsilon^2)\Big).$$

Thus with high probability, R lies in the thin spherical shell

$$R \in \big[(1-\varepsilon)\rho^\star,\; (1+\varepsilon)\rho^\star\big],$$

whose relative thickness is $O(1/\sqrt{n})$.

Connection to SSD. For a Gaussian prior $\mathcal{N}(\mathbf{0}, \sigma_{\text{gauss}}^2 I_{3n})$ (i.e., $\sigma_{\text{gauss}} = \sigma_T$ in the diffusion setting), the typical radius is $\rho^\star = \sigma_{\text{gauss}}\sqrt{3n}$, determined solely by dimension and variance. By contrast, SSD explicitly initializes on a spherical shell of radius r matched to dataset-level molecular scales, and its forward noise only broadens this δ-shell into a thin layer of thickness τ. Hence SSD replaces the *mismatched Gaussian shell* with a *geometry-aware shell* aligned to molecular radii, while structured reverse dynamics further regulate thickness.

Takeaway. An isotropic Gaussian in $3n$ dimensions behaves like a random Frobenius-norm $R = \|\mathbf{X}\|_2$ tightly concentrated around $\rho^\star = \sigma_{\text{gauss}}\sqrt{3n}$. We emphasize that this concentration applies to the *joint* (Frobenius) norm. Three scales must be distinguished: (i) per-atom 3D distance under $\mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I}_3)$, which has mean $\sigma_T\sqrt{3}$ and is n-independent; (ii) the joint Frobenius scale $\sigma_T\sqrt{3n}$;

and (iii) the molecule's own Frobenius scale $r_{\text{chem}}\sqrt{n}$, where r_{chem} is the per-atom mean radius of centered conformations.

A faithful prior should match scale (iii). The Gaussian prior matches it only when $\sigma_T\sqrt{3} = r_{\text{chem}}$, which is generally false: the noise schedule σ_T is chosen for SDE stability while r_{chem} is set by molecular geometry, so the two are not aligned in general. The per-atom mismatch $\Delta_{\text{atom}} = |\sigma_T\sqrt{3} - r_{\text{chem}}|$ and the relative error $\Delta_{\text{atom}}/r_{\text{chem}}$ do not depend on n. What grows is the absolute discrepancy: the joint mismatch is $\sqrt{n}\,\Delta_{\text{atom}}$, so the reverse process must make absolute corrections that grow with molecule size, even though the relative mismatch does not. SSD eliminates the mismatch at its source by initializing each atom directly at the chemically calibrated radius $r_{\text{SSD}} \approx r_{\text{chem}}$, so that both per-atom and joint Frobenius scales align with realistic molecular geometry. Structured dynamics (repulsion and thin-shell noise) then maintain this alignment throughout the generative process.

C Detailed Descriptions of Baseline Models

Conditional Generation Models

- **GeoDiff** [Xu et al., 2022]: GeoDiff formulates molecular conformation generation as a denoising diffusion process on atomic coordinates. The model uses an SE(3)-equivariant graph neural network to model the reverse denoising function, ensuring rotational and translational symmetry. Specifically, it implements message passing on invariant features and predicts directional updates via an equivariant edge-to-node transformation. During the forward process, Gaussian noise is added to atomic positions, and the network learns to reverse this process given the molecular graph.
- **ConfGF** [Shi et al., 2021]: ConfGF uses conditional Score Matching to model the gradient of the data distribution. Rather than relying on a parameterized diffusion kernel, it directly learns score functions over 3D coordinates using a force field-like design. The model is based on continuous-filter convolution (similar to SchNet [Schütt et al., 2017]) and employs Langevin dynamics to iteratively refine generated structures. It conditions on the molecular graph and guides generation toward conformations that satisfy chemical constraints.
- **SubGDiff** [Zhang et al., 2024]: SubGDiff enhances molecular representation learning by introducing substructure-aware diffusion. Instead of injecting noise uniformly across all atoms, it adds noise selectively to subgraphs using a discrete mask distribution. The model also includes three novel training mechanisms: (i) a subgraph prediction loss to guide structure-aware denoising, (ii) expectation state diffusion to simplify the forward process, and (iii) k-step same-subgraph diffusion to improve convergence. Its denoising network uses a standard GNN architecture without explicit geometric equivariance.
- **CGCF** [Xu et al., 2021a]: CGCF (Conditional Graph Conformation Flow) combines graph neural networks with flow-based modeling for conditional conformation generation. It conditions on the 2D molecular graph and learns invertible flows that map Gaussian priors to valid 3D conformations, enforcing chemical validity during decoding.
- **ConfVAE** [Xu et al., 2021b]: ConfVAE adopts a variational autoencoder framework for conformation generation. It encodes 2D molecular graphs into latent variables and decodes them into 3D conformations using a graph-conditioned decoder. The method emphasizes diversity in conformer sampling via latent variable regularization.
- **GeoMol** [Ganea et al., 2021]: GeoMol builds conformations by explicitly predicting torsional angles conditioned on the molecular graph. By combining learned torsional distributions with geometric constraints, GeoMol directly generates chemically valid conformations without iterative refinement.
- **Tor. Diff. (Torsional Diffusion)** [Jing et al., 2022]: Torsional Diffusion performs conditional 3D generation directly in the space of torsional angles rather than Cartesian coordinates. The model applies diffusion to rotatable bond angles while enforcing rigid-body constraints on local ring systems. A torsion-conditioned SE(3)-equivariant decoder then reconstructs Cartesian coordinates. Operating in torsion space reduces geometric redundancy and often yields strong performance on flexible molecules.

- **MCF (Molecular Conformation Fields)** [Wang et al., 2024]: MCF formulates conformation generation as a field-based prediction problem. Instead of iteratively denoising coordinates, it predicts continuous spatial potential fields that guide atoms toward chemically plausible positions. MCF incorporates torsion-aware graph message passing and Gaussian mixture fields to model multimodal conformations. It achieves strong performance on GEOM-QM9 and GEOM-Drugs in its default Gaussian-based formulation.

Unconditional Refinement Models

- **EDM (Equivariant Diffusion Model)** [Hoogeboom et al., 2022]: EDM applies denoising Score Matching in SE(3)-equivariant latent spaces. It uses the Equivariant Graph Neural Network (EGNN) architecture to perform message passing over both atomic features and coordinates, preserving SE(3) symmetry. EDM perturbs 3D coordinates using Gaussian prior noise and learns to predict score vectors that denoise atom positions. The model operates directly on 3D atom clouds and serves as a standard baseline for unconditional conformation generation.
- **GCDM (Generative Conformation Diffusion Model)** [Morehead and Cheng, 2024]: GCDM extends diffusion modeling by simultaneously reconstructing both atomic coordinates and pairwise distance matrices. It uses an SE(3)-equivariant network similar to EGNN, and introduces auxiliary distance-based losses to improve structural realism. GCDM supports both conditional and unconditional refinement modes through configurable graph encoders.
- **MuDiff** [Hua et al., 2024]: MuDiff introduces a variational diffusion framework that supports diverse conformer generation for a given molecule. It models latent variables via amortized variational inference, and performs conformation sampling using conditional Langevin dynamics guided by an energy-based score network. Its score network is based on an SE(3)-equivariant architecture adapted from EGNN.
- **GeoLDM** [Xu et al., 2023]: GeoLDM applies latent diffusion modeling to molecular structures. Instead of diffusing directly in coordinate space, it encodes molecules into a lower-dimensional latent space where Gaussian noise is added, and then decodes back into 3D coordinates. This design reduces computational cost while maintaining structural fidelity.
- **EQGAT-diff** [Le et al., 2024]: EQGAT-diff employs an equivariant graph attention network for molecular refinement. It incorporates attention-based message passing over both scalar and vector features, enabling efficient modeling of long-range dependencies. The diffusion process follows standard Gaussian corruption and denoising, while the EGAT architecture improves expressiveness.
- **FlowMol** [Dunn and Koes, 2024]: FlowMol adopts a flow-matching framework for unconditional molecular conformation generation. It learns a continuous transport map between Gaussian priors and molecular conformations using neural ODEs. FlowMol combines graph-based encoders with equivariant coordinate decoders, achieving fast generation without iterative denoising.
- **SemlaFlow** [Irwin et al., 2025]: SemlaFlow is a flow-based generative model that avoids iterative denoising by modeling a bijective mapping between latent and 3D coordinate space. It combines graph message passing and equivariant transformation modules, incorporating multi-head latent attention. Unlike diffusion-based models, SemlaFlow directly generates atomic coordinates, atom types, and charges from latent variables using Flow Matching.

These models form a representative suite spanning diffusion-based, variational, and flow-based generative frameworks. They cover both conditional (2D→3D) and unconditional (3D refinement) tasks, and employ a range of architectures from SE(3)-equivariant GNNs to transformer-based attention networks.

D Details of Evaluation Metrics

Summary. We follow the established evaluation protocols used in prior work. For *conditional generation*, we adopt the SubGDiff protocol [Zhang et al., 2024]. For *unconditional refinement*,

we adopt the SemlaFlow protocol [Irwin et al., 2025]. All thresholds, alignment steps, and post-processing procedures match the original implementations to ensure strictly comparable results.

Conditional Generation Metrics. Following Zhang et al. [2024], generated conformations are compared against ground-truth references using:

- **Coverage (COV-R/P, ↑)** — fraction of reference (R) or predicted (P) conformers matched within the RMSD threshold (0.5 Å on QM9; 1.25 Å on Drugs).
- **Matching (MAT-R/P, ↓)** — RMSD between each reference or predicted conformer and its closest counterpart in the opposite set.

Unconditional Refinement Metrics. Following Irwin et al. [2025]:

- **Atom/Molecule Stability (↑)** — percentage of stable atoms or molecules under a 1 Å neighborhood test.
- **Validity (↑)** — percentage of chemically valid conformations (RDKit sanitization).
- **Uniqueness (↑)** — fraction of distinct conformers after duplicate removal.

MCF Evaluation Protocol. To assess robustness across evaluation settings, we also evaluate SSD on the Conformer Fields (MCF) backbone [Wang et al., 2024] under its *original* GEOM-QM9 protocol. This protocol uses:

- the official GEOM-QM9 split from Ganea et al. [2021];
- the COV and AMR (average minimal RMSD) metrics used in Jing et al. [2022], Ganea et al. [2021];
- an RMSD threshold of 0.5 Å for COV;
- the **DDPM sampler with 1000 steps** as the default evaluation setting (MCF Table 1).

Our SSD-MCF results in Table 1 strictly follow this protocol without modification. We additionally report DDIM-50 results in Appendix G for completeness.

E Details of Datasets

GEOM-QM9 [Ramakrishnan et al., 2014] and **GEOM-Drugs** [Axelrod and Gomez-Bombarelli, 2022] are among the most widely adopted datasets for evaluating molecular conformation generation, particularly in both Conditional Generation and Unconditional Refinement settings.

GEOM-QM9 is a subset of the original QM9 dataset, consisting of small organic molecules with up to 9 heavy atoms (C, O, N, F). Following prior works such as GeoDiff and ConfGF, standard practice selects 40,000 molecules, each with their 5 lowest-energy conformers, resulting in approximately 200,000 training samples. The test set typically contains 200 distinct molecules and around 22,000 total conformers. These settings are designed to balance diversity and computational tractability, while enabling consistent evaluation across models.

GEOM-Drugs contains significantly more complex and diverse molecules derived from drug-like compounds. It includes around 430,000 unique molecules, with an average of 44 heavy atoms (up to 181 atoms including hydrogens). Each molecule is annotated with multiple low-energy conformers generated via energy minimization. Most benchmarks use the top 30 conformers per molecule to construct training sets. This dataset poses greater challenges for generative models due to its structural variability and conformational flexibility.

F Coordinate Updates in SE(3)-equivariant architectures

SSD is compatible with standard SE(3)-equivariant diffusion backbones. In this work, we instantiate it with GeoDiff/SubGDiff [Xu et al., 2022, Zhang et al., 2024] for conditional generation and EDM [Hoogeboom et al., 2022] (EGNN [Satorras et al., 2021]) for unconditional refinement. Although implementations differ, all backbones share the same principle: coordinate updates are expressed via inter-atomic distances and relative position vectors, ensuring SE(3)-equivariance.

GeoDiff. GeoDiff predicts edge-level scalar gradients in the SE(3)-invariant distance space and maps them back to coordinates via an equivariant transform:

$$\mathbf{g}_t^{(i)} \;=\; \sum_{j\neq i} s_{ij}\,\frac{\mathbf{x}_t^{(i)}-\mathbf{x}_t^{(j)}}{\|\mathbf{x}_t^{(i)}-\mathbf{x}_t^{(j)}\|}, \quad s_{ij} \;=\; \frac{\partial \log q}{\partial d_{ij}}, \quad d_{ij} \;=\; \|\mathbf{x}_t^{(i)}-\mathbf{x}_t^{(j)}\|.$$

Since the update depends only on normalized relative displacements and distance-invariant scalars, it is equivariant to rotations and translations.

SubGDiff. SubGDiff [Zhang et al., 2024] retains GeoDiff's coordinate update rule and adds subgraph-level masking and prediction modules on top; its SE(3)-equivariance follows directly from GeoDiff.

EDM (EGNN). EDM adopts the Equivariant Graph Neural Network (EGNN) [Satorras et al., 2021], which integrates scalar-to-vector mapping into message passing:

$$\mathbf{x}_t^{(i)} \;\leftarrow\; \mathbf{x}_t^{(i)} \;+\; \sum_{j\neq i} \big(\mathbf{x}_t^{(i)}-\mathbf{x}_t^{(j)}\big)\,\phi_x\big(m_{ij}\big),$$

where m_{ij} encodes node/edge features and (invariant) distance statistics (e.g., d_{ij}^2), and ϕ_x outputs a rotation-invariant scalar. This yields the same effect as GeoDiff's scalar-to-gradient chain rule without explicit Jacobian computation. EGNN's SE(3)-equivariance is formally proven in [Satorras et al., 2021].

SSD with Flow Matching. Many recent flow-matching methods for molecules operate in latent space (e.g., SemlaFlow [Irwin et al., 2025]). In SSD-Flow, we instead apply Flow Matching directly in coordinate space: we retain SSD's spherical-shell initialization and radial dynamics, and replace score matching with a flow-matching loss [Lipman et al., 2023]. For features, we borrow substructure-aware encodings (e.g., subgraph pooling and attention), while coordinate updates remain EGNN-parameterized to preserve SE(3)-equivariance.

Our usage. When instantiating SSD with GeoDiff, SubGDiff, or EDM, we leave their coordinate update rules unchanged and simply substitute SSD dynamics for Gaussian dynamics. For Flow Matching, we follow the same coordinate-space formulation while borrowing feature encodings from prior work. Thus, SE(3)-equivariance is inherited from the underlying equivariant networks, while SSD contributes the geometry-aware initialization and dynamics.

G Extended Baseline Comparisons

This appendix supplements the main results in Table 1 by providing full comparisons against additional molecular generation baselines. The main paper focuses on evaluating SSD as a plug-in improvement to five representative backbones, showing paired improvements under identical training and sampling budgets. Beyond these core backbones, the broader literature includes a wide range of diffusion-, equivariant-, and flow-based architectures. For completeness, we report extended conditional and unconditional results in the following sections, allowing a more comprehensive view of how SSD-enhanced models compare against earlier baselines.

G.1 Conditional Generation (Extended)

Table 6 reports extended conditional-generation results under the SubGDiff evaluation protocol. In addition to GeoDiff and SubGDiff, we include several earlier diffusion and VAE-based generators (CGCF [Xu et al., 2021a], ConfVAE [Xu et al., 2021b], GeoMol [Ganea et al., 2021], Torsional Diffusion [Jing et al., 2022], and MCF [Wang et al., 2024]) to provide broader context. These baselines span a wide range of modeling approaches and performance levels; indeed, several of them outperform the original GeoDiff backbone in their native form. The purpose of this extended comparison is not head-to-head benchmarking across heterogeneous architectures, but rather to illustrate how SSD upgrades a weaker coordinate-based backbone when evaluated under the same protocol. Consistent with this goal, SSD-GeoDiff achieves substantial improvements over its backbone and attains accuracy that is competitive with, or stronger than, these earlier baselines under the SubGDiff protocol.

Table 6: Extended conditional generation results (full baseline comparison). SubGDiff is omitted here since it is already fully analyzed in the main text. The focus of this appendix table is to illustrate that although GeoDiff is weaker than several prior baselines (e.g., Tor. Diff., MCF), SSD-GeoDiff* surpasses all of them. * indicates cases where SSD lifts a weaker backbone to outperform stronger earlier baselines.

Dataset	Model	Steps	COV-R ↑		COV-P ↑		MAT-R ↓		MAT-P ↓	
			mean	median	mean	median	mean	median	mean	median
QM9	CGCF	–	78.05	82.48	36.49	33.57	0.4219	0.3900	0.6615	0.6427
	ConfVAE	–	77.84	88.20	38.02	34.67	0.4154	0.3739	0.6215	0.6091
	GeoMol	–	71.26	72.00	–	–	0.3731	0.3731	–	–
	GeoDiff	5000	80.36	83.82	53.66	50.85	0.2820	0.2799	0.6673	0.4214
	Tor. Diff.	5000	84.26	84.75	54.91	53.47	0.2847	0.2803	0.6582	0.4203
	MCF	5000	87.14	88.36	55.83	54.68	0.2649	0.2711	0.6493	0.4538
	SSD-GeoDiff*	5000	**92.00**	**93.80**	**61.40**	**59.75**	**0.2425**	**0.2508**	**0.4188**	**0.4136**
Drugs	CGCF	–	53.96	57.06	21.68	13.72	1.2487	1.2247	1.8571	1.8066
	ConfVAE	–	55.20	59.43	22.96	14.05	1.2380	1.1417	1.8287	1.8159
	GeoMol	–	67.16	71.71	–	–	1.0875	1.0586	–	–
	GeoDiff	500	64.12	75.56	52.79	50.29	–	–	–	–
	Tor. Diff.	500	72.10	77.78	66.88	64.41	1.0532	1.0364	1.4725	1.4429
	MCF	500	74.45	81.33	64.97	62.29	1.0425	1.0267	1.5284	1.4905
	SSD-GeoDiff*	500	**91.69**	**94.86**	**70.23**	**68.11**	**0.8463**	**0.8247**	**1.1543**	**1.0894**

G.2 Unconditional Refinement (Extended)

Table 7 reports extended unconditional-refinement results under the SemlaFlow evaluation protocol, including prior diffusion-, ODE-, and flow-based refinement models such as GCDM [Morehead and Cheng, 2024], FlowMol [Dunn and Koes, 2024], GeoLDM [Xu et al., 2023], and MuDiff [Hua et al., 2024]. These baselines differ substantially in architecture and design philosophy, and several are known to outperform the original EDM backbone—particularly on the more challenging GEOM-Drugs benchmark. The goal of this extended comparison is to assess whether SSD can strengthen a weaker coordinate-space refinement model when evaluated under the same protocol. When SSD is applied to EDM, we observe large and consistent gains in stability, validity, and uniqueness. In many settings, SSD-EDM reaches performance on par with or exceeding stronger earlier baselines included for context, demonstrating that shell-guided dynamics substantially improve coordinate-space refinement even without architectural changes.

Table 7: Extended unconditional refinement results (full baseline comparison). EDM is noticeably weaker than several prior refinement models, but SSD-EDM* substantially improves stability and validity under the same protocol. * indicates cases where SSD lifts a weaker backbone to match or surpass stronger earlier baselines.

Dataset	Model	Step	Atom Stab↑	Mol Stab↑	Valid↑	Unique↑
QM9	GCDM	1000	98.7	85.7	94.8	98.4
	FlowMol	100	99.7	96.2	97.3	–
	GeoLDM	1000	98.9	89.4	93.8	98.8
	MuDiff	1000	98.8	89.9	95.3	99.1
	EDM	1000	98.7	82.0	91.9	98.9
	SSD-EDM*	1000	**99.3**	**99.8**	**100.0**	**100.0**
Drugs	GCDM	1000	89.0	5.2	–	–
	FlowMol	100	99.0	67.5	51.2	–
	GeoLDM	1000	98.9	61.5	99.3	–
	MuDiff	1000	84.0	60.9	98.9	–
	EDM	1000	81.3	–	–	–
	SSD-EDM*	1000	**90.5**	**70.1**	**98.5**	99.2

H Analysis of Shell Thickness

Notation. As in Section 3, we use $r = r_{\text{SSD}}$ to denote the fixed SSD shell radius. The shell thickness τ quantifies the radial spread of the forward distribution around this radius, accumulated from two distinct sources: stochastic broadening from the forward Brownian term $\sigma_t \, d\mathbf{w}_t$, and a geometric residual from centering an independently shell-sampled configuration to zero center of mass.

Deterministic limit. In the absence of noise ($\sigma_t = 0$), the normalized directional drift (Eq. 1) contracts each atom's distance to its assigned shell point linearly at rate α_t, yielding a delta-shell at radius $r = r_{\text{SSD}}$ once the schedule satisfies $\int_0^T \alpha_t \, dt \geq \max_i d_0^{(i)}$. The formal derivation is provided in Appendix J.1.

Source 1: Stochastic broadening. With small additive noise $\sigma_t \, d\mathbf{w}_t$, Brownian motion introduces radial variance bounded by

$$\text{Var}[\delta_T^{\text{Brownian}}] \;\leq\; \int_0^T \sigma_t^2 \, dt, \qquad \tau_{\text{Brownian}} \;\approx\; \sqrt{\int_0^T \sigma_t^2 \, dt}. \tag{5}$$

For SSD's thin-shell noise schedule used in coordinate diffusion backbones ($\sigma_t \sim 10^{-6}$ to 10^{-4}, $T \sim 1000$), $\tau_{\text{Brownian}} \sim 10^{-3}$ to 10^{-2} Å, negligible compared with r. In the implemented corruption path (Appendix J.5, item C5), this noise is added directly around the deterministic path, $\mathbf{x}_t = \bar{\mathbf{x}}_t + \tau_t \boldsymbol{\epsilon}$, rather than accumulated by simulating the SDE.

Note on EDM's noise schedule. EDM's native noise schedule has a much larger $\sigma_{\max} \approx 0.5$, used to define the score network's input conditioning and the standard EDM perturbation kernel for the score target (Appendix K). This is conceptually distinct from the small thin-shell σ_t that controls SSD's forward SDE path width. In our SSD-EDM training, EDM's larger schedule parametrises the score-target reference distribution (the thin-shell Gaussian approximation around $\bar{\mathbf{x}}_t$), while the SSD path itself remains tightly concentrated around the deterministic shell trajectory; the two schedules co-exist without conflict.

Source 2: Post-centering geometric residual. A second, generally larger contribution to τ comes from centering. As described in Section 3, an independently shell-sampled configuration $\{\mathbf{x}_T^{(i)}\}_{i=1}^n$ has a non-zero empirical center of mass; re-centering it to remove translations causes individual atomic radii to deviate from r by $O(r/\sqrt{n})$. For $n \sim 20$–50 typical of QM9 / Drugs, this geometric residual is approximately 0.9–1.5 Å on QM9 ($r = 6.37$ Å), several orders of magnitude larger than the Brownian contribution, and therefore dominates the effective shell thickness τ in practice.

Combined thickness. The combined effective thickness is

$$\tau \;\leq\; \tau_{\text{Brownian}} + r/\sqrt{n}, \tag{6}$$

which on QM9 evaluates to roughly 0.9–1.5 Å, giving a relative thickness $\tau/r \leq 0.25$. Empirically, the shell-aware reverse dynamics actively contract trajectories back toward r during sampling (Table 5). Crucially, the same radial attraction $\mathbf{v}_{\text{rad}}$ that absorbs Brownian broadening also absorbs the centering residual; no additional mechanism is required.

Role of τ. τ thus serves two compatible roles: (i) it ensures a non-singular forward SDE by injecting minimal stochastic support around the deterministic shell trajectory, and (ii) it accommodates the unavoidable $O(r/\sqrt{n})$ geometric offset introduced by centering, without altering the qualitative shell geometry that drives SSD's empirical gains.

I SE(3)-Equivariance of SSD

We show that SSD is SE(3)-equivariant *up to global translations*, i.e., equivariant on the quotient space $\mathbb{R}^{n\times 3}/\text{Transl}$ typically used in molecular modeling. For a rigid motion $g(\mathbf{x}) = R\mathbf{x} + \mathbf{t}$ with $R \in \text{SO}(3)$ acting nodewise on $\mathbf{X} = \{\mathbf{x}^{(i)}\}$, let the centering operator be

$$C(\mathbf{X}) = \mathbf{X} - \mathbf{1}\mu(\mathbf{X})^\top, \qquad \mu(\mathbf{X}) = \tfrac{1}{n}\sum_i \mathbf{x}^{(i)}.$$

Centering removes translations and commutes with rotations, since $C(g\mathbf{X}) = RC(\mathbf{X})$. We prove that the centered SSD dynamics satisfy

$$C(F(g\mathbf{X})) = R\,C(F(\mathbf{X})), \tag{7}$$

which establishes SE(3)-equivariance modulo global translations.

I.1 Forward Dynamics

The forward drift in Eq. (1) directs each atom toward its assigned shell point $\mathbf{x}_T^{(\pi(i))}$:

$$f^{(i)}(\mathbf{x}_t, t) = \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|}.$$

To establish SE(3)-equivariance, we show that the drift transforms covariantly under rigid motions applied to the entire system configuration.

Rotational covariance of the drift. Consider a rigid rotation $g(\mathbf{x}) = R\mathbf{x}$. Applying this rotation to both the current positions $\mathbf{x}_t$ and the target shell configuration $\mathbf{x}_T$, the drift transforms as follows:

$$\begin{aligned}
f^{(i)}(R\mathbf{x}_t, t) &= \alpha_t \frac{R\mathbf{x}_T^{(\pi(i))} - R\mathbf{x}_t^{(i)}}{\|R\mathbf{x}_T^{(\pi(i))} - R\mathbf{x}_t^{(i)}\|} && (8)\\
&= \alpha_t \frac{R(\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)})}{\|R(\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)})\|} && (9)\\
&= \alpha_t \frac{R(\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)})}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|} \quad \text{(rotation preserves norm)} && (10)\\
&= R \cdot \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|} && (11)\\
&= R\, f^{(i)}(\mathbf{x}_t, t). && (12)
\end{aligned}$$

This confirms that the drift is SE(3)-equivariant with respect to the system state $(\mathbf{x}_t, \mathbf{x}_T)$.

Rotational invariance of the shell initialization. The target configuration is sampled from the shell prior:

$$\mathbf{x}_T^{(i)} = r_{\text{SSD}} \frac{\mathbf{v}^{(i)}}{\|\mathbf{v}^{(i)}\|}, \qquad \mathbf{v}^{(i)} \sim \mathcal{N}(0, I). \tag{13}$$

Since the isotropic Gaussian distribution is rotationally invariant, the target configuration $\{\mathbf{x}_T^{(i)}\}$ is rotationally invariant in distribution:

$$\{R\mathbf{x}_T^{(i)}\}_{i=1}^n \overset{d}{=} \{\mathbf{x}_T^{(i)}\}_{i=1}^n, \qquad \forall R \in \text{SO}(3). \tag{14}$$

Centered forward dynamics. Let $\tilde{\mathbf{X}} = C(\mathbf{X})$ and $\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \mu(\mathbf{X})$ denote centered coordinates. Under a rigid motion g, the centered drift satisfies:

$$\tilde{f}^{(i)}(g\mathbf{X}, t) = C(f^{(i)}(g\mathbf{X}, t)) = C(R\, f^{(i)}(\mathbf{X}, t)) = R\, C(f^{(i)}(\mathbf{X}, t)) = R\, \tilde{f}^{(i)}(\mathbf{X}, t),$$

where we use the fact that centering commutes with rotation: $C(R\mathbf{X}) = RC(\mathbf{X})$.

Brownian motion invariance. Isotropic Brownian motion $d\mathbf{w}_t$ is both rotation- and translation-invariant by definition:

$$Rd\mathbf{w}_t \overset{d}{=} d\mathbf{w}_t, \qquad \forall R \in \text{SO}(3). \tag{15}$$

Combining the equivariant drift with rotationally invariant noise and applying centering after each infinitesimal update yields:

$$C(d(g\mathbf{x}_t^{(i)})) = R\, C(d\mathbf{x}_t^{(i)}),$$

establishing Eq. (7) for the forward SDE.

I.2 Reverse Dynamics

The SSD reverse SDE combines (i) inward radial attraction, (ii) short-range repulsion based only on pairwise distances and displacements, and (iii) an SE(3)-equivariant score network s_θ operating on centered coordinates:

$$d\mathbf{x}_t^{(i)} = \Big[-\alpha_t \frac{\tilde{\mathbf{x}}^{(i)}}{\|\tilde{\mathbf{x}}^{(i)}\|} + \alpha_t \, \mathbf{F}_{\text{rep}}^{(i)}(\tilde{\mathbf{X}}_t) + \sigma_t^2 \, s_\theta(\tilde{\mathbf{X}}_t, t)^{(i)} \Big] dt + \sigma_t \, d\bar{\mathbf{w}}_t^{(i)}.$$

We verify that each component is SE(3)-equivariant on centered coordinates:

(i) Radial attraction. The term $-\alpha_t \tilde{\mathbf{x}}^{(i)}/\|\tilde{\mathbf{x}}^{(i)}\|$ depends only on the normalized position vector, which transforms as:

$$\frac{R\tilde{\mathbf{x}}^{(i)}}{\|R\tilde{\mathbf{x}}^{(i)}\|} = \frac{R\tilde{\mathbf{x}}^{(i)}}{\|\tilde{\mathbf{x}}^{(i)}\|} \quad \text{(rotation preserves norm)} \tag{16}$$

$$= R \, \frac{\tilde{\mathbf{x}}^{(i)}}{\|\tilde{\mathbf{x}}^{(i)}\|}. \tag{17}$$

Therefore, the radial attraction transforms covariantly:

$$-\alpha_t \frac{R\tilde{\mathbf{x}}^{(i)}}{\|R\tilde{\mathbf{x}}^{(i)}\|} = R \cdot \left(-\alpha_t \frac{\tilde{\mathbf{x}}^{(i)}}{\|\tilde{\mathbf{x}}^{(i)}\|} \right). \tag{18}$$

(ii) Short-range repulsion. The repulsion term, restated from Eq. (2) in centered coordinates, is:

$$\mathbf{F}_{\text{rep}}^{(i)}(\tilde{\mathbf{X}}) = \alpha_t \sum_{j \neq i} \mathbb{I}(d_{ij} < d_{\min}) \, \frac{d_{\min} - d_{ij}}{d_{\min}} \, \frac{\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)}}{d_{ij}}, \tag{19}$$

where $d_{ij} = \|\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)}\|$.

The repulsion term depends on:

- **Pairwise distances** d_{ij}, which are rotation-invariant:

$$\|R\tilde{\mathbf{x}}^{(i)} - R\tilde{\mathbf{x}}^{(j)}\| = \|R(\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)})\| = \|\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)}\|. \tag{20}$$

- **Normalized displacement vectors** $(\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)})/d_{ij}$, which are rotation-covariant unit vectors: rotation preserves both numerator direction and denominator magnitude, so the quotient transforms as $R(\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)})/d_{ij}$.

Thus, under rotation:

$$\mathbf{F}_{\text{rep}}^{(i)}(C(R\mathbf{X})) = \mathbf{F}_{\text{rep}}^{(i)}(RC(\mathbf{X})) \tag{21}$$

$$= \alpha_t \sum_{j \neq i} \mathbb{I}(d_{ij} < d_{\min}) \, \frac{d_{\min} - d_{ij}}{d_{\min}} \, \frac{R(\tilde{\mathbf{x}}^{(i)} - \tilde{\mathbf{x}}^{(j)})}{d_{ij}} \tag{22}$$

$$= R \, \mathbf{F}_{\text{rep}}^{(i)}(C(\mathbf{X})). \tag{23}$$

(iii) Score correction. By construction, the score network s_θ is SE(3)-equivariant. This property is inherited from the backbone architectures such as EGNN [Satorras et al., 2021], GeoDiff [Xu et al., 2022], and SubGDiff [Zhang et al., 2024], which are designed to satisfy:

$$s_\theta(C(R\mathbf{X}), t) = R \, s_\theta(C(\mathbf{X}), t), \qquad \forall R \in \text{SO}(3). \tag{24}$$

The equivariance of EGNN follows from its coordinate update rule:

$$\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + \sum_{j \neq i} (\mathbf{x}^{(i)} - \mathbf{x}^{(j)}) \phi_x(m_{ij}), \tag{25}$$

where m_{ij} encodes rotation-invariant features (distances, angles) and ϕ_x outputs a scalar. This construction ensures coordinate updates transform covariantly under rotations, as proven in Satorras et al. [2021].

Similarly, GeoDiff and SubGDiff use edge-level score predictions in distance space, which are then mapped to coordinates via:

$$\mathbf{g}^{(i)} = \sum_{j \neq i} s_{ij} \frac{\mathbf{x}^{(i)} - \mathbf{x}^{(j)}}{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|}, \tag{26}$$

where s_{ij} depends only on invariant quantities. This formulation is manifestly SE(3)-equivariant.

I.3 Combined Reverse SDE

Combining all three components and noting that isotropic Brownian motion is rotationally invariant, we obtain:

$$C(f(g\mathbf{X}, t)) = C\Big(R\,\mathbf{v}_{\text{rad}}(C(\mathbf{X}), t) + R\,\mathbf{v}_{\text{rep}}(C(\mathbf{X}), t) + R\,\mathbf{v}_{\text{score}}(C(\mathbf{X}), t)\Big) \tag{27}$$

$$= R\,C\Big(\mathbf{v}_{\text{rad}}(C(\mathbf{X}), t) + \mathbf{v}_{\text{rep}}(C(\mathbf{X}), t) + \mathbf{v}_{\text{score}}(C(\mathbf{X}), t)\Big) \tag{28}$$

$$= R\,C(f(\mathbf{X}, t)), \tag{29}$$

where we use the fact that centering is linear and commutes with rotation.

The same property holds for the associated probability-flow ODE, which is obtained by setting the stochastic term $\sigma_t d\bar{\mathbf{w}}_t$ to zero.

I.4 Conclusion

Since centering removes translations and commutes with rotations, and all SSD drift and noise terms transform covariantly in the centered frame, Eq. (7) holds for both the forward and reverse SDEs. Hence SSD is SE(3)-equivariant on $\mathbb{R}^{n\times 3}/\text{Transl}$, which matches the natural equivalence class for molecular conformations and the evaluation protocols used in all benchmarks.

Practical verification. The SE(3)-equivariance of SSD can be verified empirically by:

1. Generating a sample $\mathbf{x}_0$ from the model
2. Applying a random rotation $R \sim \text{SO}(3)$ to obtain $R\mathbf{x}_0$
3. Rerunning generation with the same random seed but rotated initialization
4. Verifying that the output matches $R\mathbf{x}'_0$ where $\mathbf{x}'_0$ is the original output

We perform this test on 1000 randomly sampled molecules from GEOM-QM9 and verify that the maximum coordinate deviation is below 10^{-6} Å (numerical precision), confirming the theoretical SE(3)-equivariance established above.

J Thin-Shell Analysis and Forward–Reverse Radial Stability

This appendix proves that (i) the SSD forward SDE concentrates onto a thin shell via linear, distance-independent contraction toward the assigned shell points, and reports (ii) empirical radial stability of the reverse SDE, which is a geometry-guided hybrid sampler rather than the time reversal of the forward process. Together, these results justify the use of shell-based forward and reverse dynamics in SSD.

Notation. Throughout this appendix, $r = r_{\text{SSD}}$ denotes the fixed SSD shell radius (as defined in Section 3), calibrated to the mean radius of centered training conformations (e.g., $r = 6.37$ Å for GEOM-QM9, as listed in Table 9). We use $r_t^{(i)} = \|\mathbf{x}_t^{(i)}\|$ to denote the radial distance of atom i at time t during the diffusion process.

Forward SDE. For each atom, the forward dynamics are

$$d\mathbf{x}_t^{(i)} = \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|}\, dt + \sigma_t\, d\mathbf{w}_t^{(i)}, \qquad \sigma_t \ll 1, \tag{30}$$

where $\mathbf{x}_T^{(\pi(i))}$ is the assigned target point on the shell with $\|\mathbf{x}_T^{(\pi(i))}\| = r = r_{\text{SSD}}$. The normalized drift contracts each atom's distance to its assigned target at the uniform rate α_t, regardless of its initial value. This linear (rather than exponential) contraction is what produces the shell geometry analyzed in the next subsection.

J.1 Deterministic Shell Convergence

Analysis of the deterministic case ($\sigma_t = 0$). Consider first the deterministic limit where Brownian motion is absent:

$$\frac{d\mathbf{x}_t^{(i)}}{dt} = \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|}. \tag{31}$$

Let $d_t^{(i)} := \|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|$ denote the distance from atom i to its assigned shell point. Taking the time derivative:

$$\frac{dd_t^{(i)}}{dt} = -\frac{(\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)})^\top}{d_t^{(i)}} \cdot \frac{d\mathbf{x}_t^{(i)}}{dt} \tag{32}$$

$$= -\frac{(\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)})^\top}{d_t^{(i)}} \cdot \alpha_t \frac{\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}}{d_t^{(i)}} \tag{33}$$

$$= -\alpha_t \frac{\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|^2}{{d_t^{(i)}}^2} \tag{34}$$

$$= -\alpha_t. \tag{35}$$

This shows that the distance to the assigned target decreases at the constant rate α_t regardless of the current distance $d_t^{(i)}$. Integrating from $t = 0$ to $t = T$:

$$d_T^{(i)} = \max\Big(0,\ d_0^{(i)} - \int_0^T \alpha_t\, dt\Big), \tag{36}$$

where the $\max$ accounts for atoms that reach their target before T and are subsequently held at the shell (we discuss the measure-zero singularity at $d_t^{(i)} = 0$ in Appendix J.5, item C7). Choosing T such that $\int_0^T \alpha_t\, dt \geq \max_i d_0^{(i)}$ guarantees that *all* atoms have reached their assigned shell points by time T, yielding a **delta-shell** at radius $r = r_{\text{SSD}}$ in the deterministic limit. Crucially, while atoms with smaller $d_0^{(i)}$ arrive earlier, the linear (rather than exponential) contraction of $d_t^{(i)}$ ensures that the spread of arrival times is bounded by $\max_i d_0^{(i)}/\alpha_t$, far tighter than the unbounded exponential tails of OU convergence.

Key distinction from Ornstein–Uhlenbeck dynamics. The contraction profile in SSD is fundamentally different from OU dynamics $dr = \alpha(r_{\text{target}} - r)dt$, whose exponential solution $d(t) = d_0 e^{-\alpha t}$ never reaches zero in finite time and assigns exponentially long tails to atoms initialized far from equilibrium. SSD's normalized drift produces *linear* contraction $d_t^{(i)} = d_0^{(i)} - \int_0^t \alpha_s ds$, which (i) reaches zero in finite, predictable time, and (ii) bounds the spread of arrival times across atoms by the diameter of the initial distribution rather than allowing it to diverge exponentially.

The practical consequence is that, by terminal time T, all atoms in SSD are concentrated on a well-defined shell at radius r_{SSD}, whereas OU dynamics produce a smeared distribution whose tails reflect the heterogeneity of initial distances.

J.2 Thin-Shell Broadening via Brownian Motion

Stochastic case ($\sigma_t > 0$). With small additive noise $\sigma_t\, d\mathbf{w}_t^{(i)}$, the deterministic shell is broadened into a thin layer. Let $\delta_t^{(i)} = r_t^{(i)} - r$ denote the radial deviation from the target shell. The radial component of the Brownian motion contributes variance:

$$\mathrm{Var}[\delta_T^{(i)}] \le \int_0^T \sigma_t^2\, dt =: \tau_{\mathrm{Brownian}}^2, \tag{37}$$

yielding a Brownian-induced thickness $\tau_{\mathrm{Brownian}} = \sqrt{\int_0^T \sigma_t^2\, dt}$.

This bound holds because (i) the deterministic drift keeps trajectories contracting toward the shell along the radial direction, (ii) Brownian motion is isotropic with radial variance per dimension $\sigma_t^2/3$, and (iii) the accumulated variance integrates over the diffusion time.

Combined thickness including the centering residual. A second contribution to the shell thickness comes from the centering step described in Section 3: an independently shell-sampled configuration is projected to zero center of mass, after which individual atomic radii deviate from r by $O(r/\sqrt{n})$. This geometric residual is generally larger than the Brownian contribution and dominates τ in practice for typical molecule sizes (see Appendix H for a quantitative comparison). The combined effective thickness is therefore

$$\tau \;\le\; \tau_{\mathrm{Brownian}} + r/\sqrt{n}. \tag{38}$$

For QM9 ($r = 6.37$ Å, $n \sim 20$–30), $\tau \approx 1$–1.5 Å, giving a relative thickness $\tau/r \le 0.25$.

Forward terminal distribution. Combining deterministic shell convergence with the broadening above, the forward terminal distribution is approximately

$$p_T(\mathbf{x}) \approx \mathrm{Unif}(\mathbb{S}^2(r)) \otimes \mathcal{N}(0, \tau^2 I_{\text{radial}}), \tag{39}$$

where $\mathrm{Unif}(\mathbb{S}^2(r))$ denotes uniform distribution on the sphere of radius r, and the radial component has Gaussian spread with variance τ^2.

J.3 Reverse SDE and Empirical Radial Stability

Reverse SDE formulation. The SSD reverse SDE combines three components:

$$d\mathbf{x}_t^{(i)} = \left[\mathbf{v}_{\mathrm{rad}}^{(i)} + \mathbf{v}_{\mathrm{rep}}^{(i)} + \mathbf{v}_{\mathrm{score}}^{(i)}\right] dt + \sigma_t\, d\bar{\mathbf{w}}_t^{(i)}, \tag{40}$$

where:

- **Radial attraction:** $\mathbf{v}_{\mathrm{rad}}^{(i)} = -\alpha_t \frac{\mathbf{x}_t^{(i)}}{\|\mathbf{x}_t^{(i)}\|}$ provides inward drift at uniform radial speed, mirroring the forward directional attraction.
- **Repulsion:** $\mathbf{v}_{\mathrm{rep}}^{(i)}$ prevents atomic collapse by enforcing minimum separation $d_{\min}$.
- **Score correction:** $\mathbf{v}_{\mathrm{score}}^{(i)} = \sigma_t^2 s_\theta(\mathbf{x}_t, t)^{(i)}$ refines local molecular geometry through a learned SE(3)-equivariant network.

Radial stability (empirical). The reverse process initializes from the same shell distribution and maintains radial concentration via $\mathbf{v}_{\mathrm{rad}}$. We do not derive a bound on the divergence between the full forward and reverse marginals. What we measure is the radial deviation of sampling trajectories from the shell (Table 5); this is evidence of radial stability only, and it does not control the angular structure, inter-atom correlations, the learned-score error, or the reverse-only repulsion $\mathbf{v}_{\mathrm{rep}}$.

Practical implications. The shell-aware reverse dynamics empirically maintain radial concentration, with measured radial deviation dropping to 0.18 Å on QM9 (compared with 0.45 Å for Gaussian baselines, Table 5). The combined picture is that SSD performs generative sampling on a thin spherical shell centered at r_{SSD}, with the centering-induced finite thickness fully absorbed by the reverse radial attraction.

J.4 Comparison with Standard Diffusion Theory

The SSD framework differs from standard score-based diffusion in two key aspects:

1. Structured prior: Rather than using an unstructured Gaussian prior $p_T = \mathcal{N}(0, \sigma_T^2 I)$, SSD uses a geometrically meaningful shell prior $p_T = \mathrm{Unif}(\mathbb{S}^2(r_{\mathrm{SSD}})) \otimes \mathcal{N}(0, \tau^2)$ where r_{SSD} is calibrated to molecular scale.

2. Hybrid reverse dynamics: The reverse drift combines analytical geometric constraints ($\mathbf{v}_{\mathrm{rad}}$, $\mathbf{v}_{\mathrm{rep}}$) with learned score correction ($\mathbf{v}_{\mathrm{score}}$). This hybrid design provides complementary geometric guidance that stabilizes sampling and reduces spatial entropy compared to purely learned score-based dynamics.

Neither modification is accompanied by a derived forward–reverse consistency bound; the radial stability of the resulting sampler is measured in Appendix J.3.

J.5 Clarifications on Forward, Reverse, and Training Consistency

This subsection addresses several technical questions that may arise from the joint reading of Section 3 (Eqs. 1 and 2) and the analysis above. Each item clarifies the relationship between the proposed dynamics and standard score-based diffusion theory.

(C1) The OU description is for the radial component, not the SDE itself. The forward SDE in Eq. (1) is a normalized directional field, not an Ornstein–Uhlenbeck (OU) process. We use the OU-form expression $dr_t = \alpha_t(r - r_t)\,dt + \sigma_t\,dB_t$ in Appendix J.1 purely as a *shorthand summary* of how the radial coordinate $r_t = \|\mathbf{x}_t^{(i)}\|$ evolves under the normalized drift; it is not a literal SDE that SSD integrates. The actual SDE is a normalized vector field of constant drift magnitude α_t in $\mathbb{R}^{n\times 3}$, which produces *linear* (not exponential) contraction of the distance to the assigned target, as proven in Appendix J.1. The OU-style summary captures the right qualitative behaviour—that atoms are pulled toward r_{SSD}—but understates the geometry: SSD's contraction is distance-independent and reaches the shell in finite time, whereas a true OU process is exponential and asymptotic. The two descriptions are therefore consistent on the radial coordinate's qualitative trend but distinct as stochastic processes.

(C2) Radial stability, not trajectory symmetry. In score-based diffusion, the forward process is prescribed analytically while the reverse relies on a learned score; the two are inherently asymmetric *at the trajectory level*. We do not claim marginal consistency between the full forward and reverse processes, and no bound on their divergence is derived; what we measure is reduced radial deviation along sampling trajectories (Table 5). Trajectory-level symmetry—a point-specific reverse mirroring the point-specific forward drift—is *not* a requirement of diffusion theory and would, as discussed below, be unimplementable at inference.

(C3) The isotropic reverse drift is the only principled choice. A trajectory-symmetric reverse drift would require, at each step, knowledge of the assignment π that pairs each atom with its target shell point $\mathbf{x}_T^{(\pi(i))}$. During training, π is sampled and known. At inference, π is marginalized out and unavailable: the reverse process runs on noisy coordinates without access to the original target assignment. The isotropic radial contraction $\mathbf{v}_{\mathrm{rad}}^{(i)} = -\alpha_t\,\mathbf{x}_t^{(i)}/\|\mathbf{x}_t^{(i)}\|$ is therefore the only design that (i) mirrors the forward radial scale without requiring π, (ii) preserves SE(3)-equivariance, and (iii) avoids brittle atom-ordering dependencies. This is a deliberate design choice grounded in inference constraints, not an ad-hoc simplification.

(C4) Score scaling and reverse SDE formulation. The reverse drift uses score scaling $\sigma_t^2\,s_\theta$ rather than the $g(t)^2\,s_\theta$ that would arise from a literal time-reversal of Eq. (1). This is consistent with the SDE formulation chosen in Section 2.3, where σ_t plays the role of the diffusion coefficient and $g(t) \equiv \sigma_t$ in the SSD setting; the two coincide. The reverse drift is thus the standard score-correction term under SSD's noise schedule, combined with the analytical geometric guidance $\mathbf{v}_{\mathrm{rad}}$ and $\mathbf{v}_{\mathrm{rep}}$ that compensate for the absence of the point-specific forward drift at inference (item C3). This hybrid construction is what we mean by "shell-aware reverse dynamics" in Section 3.

(C5) Training objective under SSD forward marginals (score-based backbones). For score-based backbones (SSD-GeoDiff, SSD-SubGDiff, SSD-EDM, SSD-MCF), the training objective is denoising score matching against the SSD *forward marginals*, not the Gaussian perturbation kernel. Two design choices need to be specified explicitly:

(i) Sampling from p_t. Training samples $\mathbf{x}_t$ are drawn in closed form, without integrating Eq. (1): with a freshly sampled (re-centered) shell target $\mathbf{x}_T$ and permutation π, $\mathbf{x}_t = \bar{\mathbf{x}}_t + \tau_t \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\bar{\mathbf{x}}_t$ is the deterministic SSD trajectory (Eq. (1) with $\sigma_t = 0$; each atom moves in a straight line toward its target at speed α_t and stops on arrival) and $\tau_t^2 = \int_0^t \sigma_s^2 \, ds$. This replaces the standard Gaussian perturbation $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$.

(ii) Computing the score target. Since $p_t(\mathbf{x}_t \mid \mathbf{x}_0, \pi)$ is non-Gaussian (a mixture over the shell draw), the baseline's closed-form Gaussian score $-(\mathbf{x}_t - \mathbf{x}_0)/\sigma_t^2$ is *not* the correct target. Conditioned additionally on $\mathbf{x}_T$, however, the kernel of (i) is exactly $\mathcal{N}(\bar{\mathbf{x}}_t, \tau_t^2 \mathbf{I})$, centered on the *deterministic* SSD trajectory: the analytic target becomes $-(\mathbf{x}_t - \bar{\mathbf{x}}_t)/\tau_t^2$, the exact conditional score, and DSM on this target has the marginal score of the SSD forward process as its minimizer.

The DSM *loss form* (L^2 regression onto an analytic score target) is identical to the backbone's, and the optimizer, learning rate, and schedule are unchanged (Appendix K); the measured training cost is also unchanged (Table 18). What changes is the *distribution from which $\mathbf{x}_t$ is drawn* and the corresponding *analytic target*; both follow mechanically from the SSD forward construction. As a consequence, s_θ becomes geometry-aware from training rather than only at inference. We note that SSD-Flow uses a different training objective (flow matching on the SSD path), described separately in Appendix K; the discussion in this subsection addresses only the score-based variants.

(C6) Marginalization over the random permutation π. The permutation π is resampled per trajectory, so the score network implicitly learns the marginal score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ where $p_t(\mathbf{x}_t) = \mathbb{E}_\pi[p_t(\mathbf{x}_t \mid \pi)]$. This marginalization is what makes the inference-time reverse process well-posed without access to π, and is essential for the permutation-symmetric treatment of indistinguishable atoms. Empirically, the resulting score is robust: as reported in Section 3 (and confirmed in Table 14), reshuffled atom–shell assignments at inference yield indistinguishable performance, while deterministic nearest-distance matching—which fixes a particular π—collapses performance, confirming that the learned score correctly captures the π-marginalized distribution.

(C7) Forward-drift normalization and the discontinuity at the target. The normalized drift in Eq. (1) is undefined exactly when $\mathbf{x}_t^{(i)} = \mathbf{x}_T^{(\pi(i))}$, but this is a measure-zero event during training and has no impact on the SDE flow. In practice, the distance $\|\mathbf{x}_T^{(\pi(i))} - \mathbf{x}_t^{(i)}\|$ decreases at constant rate α_t (deterministic-limit derivation in Appendix J.1), so the trajectory reaches the target at a unique terminal time T at which the drift is no longer applied; for $t < T$ the drift is bounded by α_t and well-defined. The associated probability-flow ODE inherits the same property. We note this is analogous to the standard treatment of normalized vector fields in optimal-transport-based generative models, where the velocity field is defined up to a measure-zero singular set without affecting the flow.

(C8) Role of $\mathbf{v}_{\mathrm{rep}}$. $\mathbf{v}_{\mathrm{rep}}$ is an analytical short-range correction that prevents atomic collapse during reverse sampling but is absent from the forward SDE. No derivation in this appendix accounts for $\mathbf{v}_{\mathrm{rep}}$'s radial and tangential contribution. Empirically, however, this contribution is negligible: Table 11 (Appendix L) shows that ablating $\mathbf{v}_{\mathrm{rep}}$ entirely ("Mean shell + σ_t" row) changes COV-R by only -0.5 relative to full SSD, while still outperforming the Gaussian baseline by $+1.8$. SSD's gains over Gaussian baselines therefore rest on the shell initialization and shell-aware radial dynamics as a whole, not on $\mathbf{v}_{\mathrm{rep}}$, and its effect on performance is empirically small. We view $\mathbf{v}_{\mathrm{rep}}$ as a physically motivated guard against atomic overlap (analogous to soft-core repulsion in molecular force fields) rather than a load-bearing component of the diffusion formalism.

K Training Details

Table 8 summarizes the training protocols used for all backbones and their SSD variants. We follow the *official* implementations of GeoDiff, SubGDiff, EDM, SemlaFlow, and MCF without modifying their architectures, optimizers, schedules, or number of steps. SSD replaces only the Gaussian prior

with a spherical-shell prior and augments the forward/reverse dynamics, leaving all other backbone components unchanged. Detailed hyperparameters and SSD-specific settings appear below.

Table 8: Training protocols for conditional (Cond.) and unconditional (Uncond.) backbones. SSD variants inherit the same settings as their corresponding backbones, with only initialization and dynamics replaced by SSD. MCF is trained following its original DDPM-based protocol from Wang et al. [2024].

Model	Task	Schedule	Steps T	Batch / LR
GeoDiff / SubGDiff [Xu et al., 2022, Zhang et al., 2024]	Cond.	2M (QM9), 6M (Drugs)	5000 (QM9), 500 (Drugs)	$64, 32$ / 2×10^{-4}
SSD-GeoDiff / SSD-SubGDiff	Cond.	same	same	same
EDM [Hoogeboom et al., 2022]	Uncond.	120K (QM9), 200K (Drugs)	1000	64 / 1×10^{-4} (cosine)
SSD-EDM	Uncond.	same	same	same
SemlaFlow [Irwin et al., 2025]	Uncond. (flow)	120K (QM9), 200K (Drugs)	100	64 / 1×10^{-4}
SSD-Flow	Uncond. (flow)	same	same	same
MCF [Wang et al., 2024]	Cond.	300K (QM9), 750K (Drugs)	1000	512 / 1×10^{-4}
SSD-MCF	Cond.	same	same	same

K.1 Auto-Calibrated SSD Parameters

One-time calibration (no tuning). We perform a single-pass calibration on the training split to determine SSD-specific geometric parameters, after which all values remain fixed across datasets and backbones. No hyperparameter tuning is performed. The calibration procedure is:

1. **Shell radius r_{SSD}:** Compute the mean absolute radius of all centered training conformations:

$$r_{\text{SSD}} = \mathbb{E}_{\mathbf{x}_0 \sim \mathcal{D}_{\text{train}}} \left[\frac{1}{n} \sum_{i=1}^{n} \|\mathbf{x}_0^{(i)} - \bar{\mathbf{x}}_0\| \right], \tag{41}$$

   where $\bar{\mathbf{x}}_0 = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_0^{(i)}$ is the center of mass.

2. **Minimum separation $d_{\min}$:** The repulsion term $\mathbf{v}_{\text{rep}}$ activates only below $d_{\min}$, so this threshold should sit strictly below every valid covalent bond length in the dataset to avoid penalizing equilibrium geometries. We use

$$d_{\min} = 0.9 \text{ Å}, \tag{42}$$

   which is below the shortest realistic covalent bond lengths in QM9 / Drugs—including O–H (~ 0.96 Å) and N–H (~ 1.01 Å)—and well below the more common C–H (~ 1.07 Å) and C–C / C–O (~ 1.4–1.5 Å) bonds. The minimum pairwise atomic distance observed in the training set,

$$\min_{\mathbf{x}_0 \sim \mathcal{D}_{\text{train}}} \min_{i \neq j} \|\mathbf{x}_0^{(i)} - \mathbf{x}_0^{(j)}\|,$$

   provides an empirical reference for this choice. In practice, $\mathbf{v}_{\text{rep}}$ activates only when atoms transiently approach unphysical overlap during reverse sampling, never on equilibrium bonds; its empirical contribution to final performance is small (Table 11, -0.5 COV-R when removed), consistent with its role as a soft-core safeguard rather than a load-bearing component of the dynamics.

The numerical values used in all experiments are listed in Table 9. All other parameters (noise schedules, drift coefficients) are inherited from each backbone without modification, introducing no additional hyperparameters beyond r_{SSD} and $d_{\min}$.

Table 9: SSD-specific geometric parameters. r_{SSD} is the chemically calibrated shell radius (per-dataset training-set mean); $d_{\min}$ is the repulsion activation threshold, set below all realistic covalent bond lengths in either dataset (distances in Å).

Dataset	r_{SSD}	$d_{\min}$
QM9	6.37	0.9
Drugs	21.16	0.9

K.2 SSD Dynamics Parameters

The SSD forward and reverse processes are governed by two schedules: the drift rate α_t and the thin-shell noise σ_t. We map these directly from the backbone's original schedules to ensure compatibility and avoid introducing additional hyperparameters.

Drift rate α_t. The drift rate controls the speed of directional attraction in both the forward (Eq. 1) and reverse processes. We set α_t to follow the same schedule as the backbone's diffusion coefficient or noise level:

- **GeoDiff/SubGDiff:** α_t follows a linear schedule from $\alpha_0 = 0.0001$ to $\alpha_T = 0.02$ over $T = 5000$ (QM9) or $T = 500$ (Drugs) steps, matching the original GeoDiff noise schedule.
- **EDM:** α_t follows a cosine schedule matching EDM's noise schedule:

$$\alpha_t = \sigma_{\min} + \frac{1}{2}(\sigma_{\max} - \sigma_{\min}) \left(1 + \cos\left(\frac{t\pi}{T}\right)\right), \tag{43}$$

  with $\sigma_{\min} = 0.0001$, $\sigma_{\max} = 0.5$, and $T = 1000$.
- **SSD-Flow:** α_t follows a linear schedule from $\alpha_0 = 0.0001$ to $\alpha_T = 0.02$ over $T = 100$ flow matching steps.
- **MCF:** α_t follows the DDPM variance schedule used in the original MCF implementation, with $T = 1000$ diffusion steps.

Thin-shell noise σ_t. The thin-shell noise introduces minimal radial broadening to avoid a singular delta-shell while maintaining tight concentration. For diffusion backbones, we inherit σ_t directly from the backbone's noise schedule $g(t)$. For flow-matching, we set $\sigma_t = 0$ for deterministic sampling:

- **GeoDiff/SubGDiff:** σ_t is inherited from the backbone's $g(t)$ schedule, which follows a linear variance schedule in the original implementation. Typical values range from $\sigma_0 \approx 10^{-6}$ to $\sigma_T \approx 10^{-4}$.
- **EDM:** σ_t matches EDM's noise schedule:

$$\sigma_t = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{t/T}, \tag{44}$$

  with $\sigma_{\min} = 0.0001$ and $\sigma_{\max} = 0.5$. We note that EDM's larger $\sigma_{\max}$ parametrises the reference distribution used in the score target (the thin-shell Gaussian centered at $\bar{\mathbf{x}}_t$, see below) and is conceptually distinct from the small thin-shell σ_t that controls SSD's forward SDE path width; see Appendix H for the relationship between the two schedules.
- **SSD-Flow:** $\sigma_t = 0$ (deterministic flow matching, no stochastic term in the reverse process).
- **MCF:** σ_t is inherited from MCF's variance schedule, following the DDPM formulation with typical values $\sigma_t \sim 10^{-5}$ to 10^{-4}.

Design rationale. By inheriting α_t and σ_t from backbone schedules, SSD introduces *zero additional hyperparameters* for dynamics beyond the geometric constants r_{SSD} and $d_{\min}$. This design ensures:

1. **Compatibility:** SSD dynamics remain consistent with the backbone's original diffusion process.
2. **Simplicity:** No hyperparameter tuning is required—calibration is a one-time dataset-level computation.
3. **Generality:** The same mapping procedure applies to all coordinate-space backbones.

K.3 Conditional Generation (GeoDiff / SubGDiff / MCF)

We use the official implementations of GeoDiff [Xu et al., 2022], SubGDiff [Zhang et al., 2024], and MCF [Wang et al., 2024], following their preprocessing, data splits, and training schedules exactly. All models are trained using the same molecular graph encoders (GIN [Xu et al., 2019] for GeoDiff/SubGDiff; SchNet-style [Schütt et al., 2017] for MCF) and coordinate update mechanisms.

GeoDiff / SubGDiff.

- **Architecture:** GIN-based graph encoder with 5 layers, hidden dimension 256, and edge features encoding bond types and distances.
- **Score network:** Predicts edge-level gradients in distance space, then maps to coordinate updates via:

$$\mathbf{g}_t^{(i)} = \sum_{j \neq i} s_{ij} \frac{\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}}{\|\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}\|}, \tag{45}$$

  where $s_{ij} = \partial \log q / \partial d_{ij}$ is the predicted distance gradient.
- **Training:** Denoising score matching with L^2 loss:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t}\left[\|\mathbf{g}_\theta(\mathbf{x}_t, t) - \mathbf{g}_{\text{true}}(\mathbf{x}_t, \mathbf{x}_0, t)\|^2\right], \tag{46}$$

  where $\mathbf{g}_{\text{true}}$ is the analytic score of the Gaussian perturbation kernel.
- **Optimizer:** Adam with learning rate 2×10^{-4}, weight decay 10^{-6}, and no learning rate scheduling.
- **Batch size:** 64 for QM9, 32 for Drugs (due to larger molecular sizes).
- **Training steps:** 2M iterations for QM9, 6M for Drugs.

SSD-GeoDiff / SSD-SubGDiff. SSD variants use identical denoiser architectures, optimizers, and training schedules. The modifications are:

1. **Initialization:** Replace Gaussian sampling $\mathbf{x}_T \sim \mathcal{N}(0, \sigma_T^2 I)$ with shell sampling followed by re-centering:

$$\mathbf{x}_T^{(i)} = r_{\text{SSD}} \frac{\mathbf{v}^{(i)}}{\|\mathbf{v}^{(i)}\|}, \quad \mathbf{v}^{(i)} \sim \mathcal{N}(0, I), \qquad \mathbf{x}_T \leftarrow \mathbf{x}_T - \text{mean}(\mathbf{x}_T). \tag{47}$$

2. **Training corruption:** Replace the Gaussian perturbation of $\mathbf{x}_0$ with the SSD corruption path (Appendix J.5, item C5).
3. **Regression target:** Replace the Gaussian score target with the exact conditional score of the SSD corruption kernel (below).
4. **Reverse dynamics:** Augment the backbone's score-based update with the radial attraction and short-range repulsion described in Section 3 (with the repulsion term given by Eq. 2).

SSD Training Modification. SSD modifies only the *data corruption mechanism* and the corresponding regression target while keeping all other training components (architecture, loss function form, optimizer, schedules) identical to the baseline:

1. **Shell distribution at $t = T$:** The forward corruption path ends at the SSD shell distribution defined in Section 3 rather than $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$.
2. **Forward corruption:** Training samples $\mathbf{x}_t$ are generated in closed form around the deterministic SSD trajectory, with a freshly sampled shell target and permutation π (Appendix J.5, item C5); the SDE is not integrated. This replaces the standard Gaussian perturbation $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$.
3. **Score target.** Marginalizing over the shell draw, the conditional kernel $p_t(\mathbf{x}_t \mid \mathbf{x}_0, \pi)$ does not admit a closed-form Gaussian density and the baseline's analytic Gaussian score is not a valid target. Conditioned additionally on the shell target, the corrupted sample is $\mathbf{x}_t = \bar{\mathbf{x}}_t + \tau_t \boldsymbol{\epsilon}$, where $\bar{\mathbf{x}}_t$ is the deterministic SSD trajectory at time t (Eq. 1 with $\sigma_t = 0$), $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\tau_t^2 = \int_0^t \sigma_s^2 \, ds$. Under this conditioning, the kernel is exactly Gaussian, centered at $\bar{\mathbf{x}}_t$ with variance τ_t^2, and the corresponding analytic score $-(\mathbf{x}_t - \bar{\mathbf{x}}_t)/\tau_t^2$ serves as the DSM regression target.

   The score network thus learns the geometry-aware score of the SSD forward marginals, while the loss *form* (denoising L^2 regression onto an analytic score target) is identical to the baseline backbone. Only the trajectory the score is regressed against changes—from a straight Gaussian path to the SSD shell-bound path.

The DSM loss form, optimizer, learning rate, batch size, and training steps remain unchanged from the baseline. The only differences are the corruption distribution from which $\mathbf{x}_t$ is drawn and the corresponding analytic score target; both follow directly from the SSD corruption path, and no optimization hyperparameters are altered (measured training cost in Table 18). Critically, the score network learns geometry-aware representations *from training*, not just at inference.

MCF. MCF uses a different architecture based on continuous convolutional layers (SchNet-style) rather than message-passing GNNs:

- **Architecture:** 6-layer SchNet with cutoff radius 5 Å, hidden dimension 128, and continuous filter size 64.
- **Training:** DDPM-based training with the standard variational lower bound objective over $T = 1000$ diffusion steps.
- **Optimizer:** Adam with learning rate 1×10^{-4}, cosine learning rate schedule, and gradient clipping at norm 1.0.
- **Batch size:** 512 (using gradient accumulation over 8 microbatches of size 64).
- **Training steps:** 300K iterations for QM9, 750K for Drugs.

SSD-MCF. SSD-MCF retains all MCF architectural and training details, replacing only the prior and dynamics with SSD components as described above (shell initialization, SSD forward corruption, and geometry-aware score targets).

K.4 Unconditional Refinement (EDM Backbone)

We follow EDM [Hoogeboom et al., 2022] with an EGNN backbone [Satorras et al., 2021], keeping all hyperparameters and schedules unchanged.

EDM.

- **Architecture:** EGNN with 7 layers, hidden dimension 256, and SE(3)-equivariant coordinate updates:
$$\mathbf{x}_t^{(i)} \leftarrow \mathbf{x}_t^{(i)} + \sum_{j \neq i} (\mathbf{x}_t^{(i)} - \mathbf{x}_t^{(j)}) \phi_x(m_{ij}), \tag{48}$$
where m_{ij} encodes node/edge features and ϕ_x is a learned scalar function.
- **Training:** Denoising score matching with noise conditioning:
$$\mathcal{L} = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_t} \left[\| \mathbf{s}_\theta(\mathbf{x}_t, \sigma_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{x}_0) \|^2 \right]. \tag{49}$$
- **Optimizer:** Adam with learning rate 1×10^{-4}, cosine annealing schedule, and weight decay 10^{-12}.
- **Batch size:** 64 for both QM9 and Drugs.
- **Training steps:** 120K iterations for QM9, 200K for Drugs.
- **Sampling:** Probability-flow ODE with 1000 function evaluations using the Heun solver.

SSD-EDM. SSD-EDM applies the same training configuration but replaces the Gaussian initialization and dynamics with SSD. The EGNN architecture and all training hyperparameters remain unchanged.

SSD Training Modification. SSD-EDM replaces Gaussian noise schedules with SSD forward dynamics while preserving all other training components:

1. **Initialization:** Replace $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \mathbf{I})$ with the SSD shell sampling defined in Section 3.
2. **Forward corruption:** Draw $\mathbf{x}_t$ in closed form around the deterministic SSD trajectory (Appendix J.5, item C5) instead of the Gaussian perturbation $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}$; the SDE is not integrated.

3. **Score matching with thin-shell target.** SSD-EDM uses the same thin-shell Gaussian construction described above for SSD-SubGDiff: the DSM regression target is computed analytically as $-(\mathbf{x}_t - \bar{\mathbf{x}}_t)/\tau_t^2$, where $\bar{\mathbf{x}}_t$ is the deterministic SSD trajectory at time t and $\tau_t^2 = \int_0^t \sigma_s^2 \, ds$. The loss *form* ($\mathrm{L}^2$ regression onto an analytic score target) is identical to EDM; only the target's reference distribution changes from a Gaussian centered at $\mathbf{x}_0$ to a Gaussian centered at $\bar{\mathbf{x}}_t$.

The EGNN architecture, optimizer (Adam with learning rate 1×10^{-4}, cosine annealing), batch size (64), and training steps (120K for QM9, 200K for Drugs) remain identical to EDM, so the score network becomes geometry-aware from training, exactly as in the SSD-SubGDiff variant.

K.5 Unconditional Refinement (SSD-Flow)

SSD-Flow adapts the Flow Matching objective [Lipman et al., 2023, Irwin et al., 2025] to coordinate space while incorporating SSD's shell-based initialization and dynamics.

Architecture. SSD-Flow uses a coordinate-space architecture based on EDM's EGNN backbone, rather than the latent-space encoder used in SemlaFlow:

- **Encoder:** 5-layer EGNN with hidden dimension 128, processing atomic coordinates and types directly.
- **Substructure features:** We incorporate subgraph pooling and attention mechanisms from SemlaFlow to maintain compatibility with existing refinement protocols:

$$\mathbf{h}_{\text{sub}} = \text{Attention}(\text{Pool}(\{\mathbf{h}_i : i \in \mathcal{S}\})), \tag{50}$$

  where $\mathcal{S}$ denotes a detected substructure (e.g., rings, functional groups).
- **Flow network:** 7-layer EGNN predicting velocity fields $\mathbf{v}_\theta(\mathbf{x}_t, t)$ for coordinate-space flow matching.

This design reduces the total parameter count from 46.6M (SemlaFlow) to 28.54M (SSD-Flow) while maintaining the same flow formulation and number of function evaluations (NFEs).

Training objective. SSD-Flow trains a coordinate-space velocity field $\mathbf{v}_\theta$ via *flow matching* [Lipman et al., 2023], regressing onto the conditional vector field of the SSD-shell probability path. The forward process generates training pairs $(\mathbf{x}_0, \mathbf{x}_t)$ by sampling $\mathbf{x}_0$ from data, $\mathbf{x}_T$ from the SSD shell prior (Section 3) with a fresh permutation π, and $\mathbf{x}_t$ by integrating the deterministic SSD forward dynamics (Eq. 1 with $\sigma_t = 0$) from $\mathbf{x}_0$ toward $\mathbf{x}_T$ up to time t. The conditional vector field along this path is exactly the SSD forward drift evaluated at the simulated point:

$$\mathbf{v}_{\text{target}}(\mathbf{x}_t, \mathbf{x}_T, t) \;=\; \alpha_t \, \frac{\mathbf{x}_T^{(\pi(\cdot))} - \mathbf{x}_t}{\|\mathbf{x}_T^{(\pi(\cdot))} - \mathbf{x}_t\|}. \tag{51}$$

The flow-matching loss is then standard:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t, \mathbf{x}_0, \mathbf{x}_T, \pi}\Big[\big\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}_{\text{target}}(\mathbf{x}_t, \mathbf{x}_T, t)\big\|^2\Big]. \tag{52}$$

Relation to the marginal-preservation property of flow matching. Because the regression target is the velocity of the conditional path, the minimizer of $\mathcal{L}_{\text{FM}}$ is the marginal velocity field of the SSD path, and integrating it transports the marginals of that path [Lipman et al., 2023]; the radial alignment that $\mathbf{v}_{\text{rad}}$ enforces in the score-based reverse process is here part of the path itself. This property holds for the learned field only and assumes that the path reaches the shell by $t = T$ ($\int_0^T \alpha_t \, dt \geq \max_i d^{(i)}$). In inference, we additionally apply the short-range repulsion $\mathbf{v}_{\text{rep}}$, which is not part of the path, so the sampler used in our experiments is not covered by this property; $\mathbf{v}_{\text{rep}}$ contributes little to performance (Appendix J.5, item C8).

Training details.

- **Optimizer:** Adam with learning rate 1×10^{-4}, cosine annealing, and weight decay 10^{-6}.

- **Batch size:** 64 for both QM9 and Drugs.
- **Training steps:** 120K iterations for QM9, 200K for Drugs.
- **Sampling:** Deterministic ODE integration ($\sigma_t = 0$) with 100 NFEs using the Dopri5 adaptive solver.

SSD Training vs. Gaussian Baseline. Unlike SemlaFlow, which performs flow matching in latent space with a Gaussian prior, SSD-Flow operates in coordinate space with a shell-based path:

- **Probability path:** The conditioning path is the SSD forward trajectory toward a chemically calibrated shell point, replacing SemlaFlow's latent-space Gaussian path.
- **Target velocity:** The regression target (Eq. 51) is the instantaneous velocity of this path, computed in closed form. No additional analytical guidance terms are added to the target.
- **Velocity network:** $\mathbf{v}_\theta$ learns to match this geometry-aware path velocity, so inference is a clean ODE integration of $\mathbf{v}_\theta$ without further analytical augmentation (apart from the inference-time repulsion $\mathbf{v}_{\text{rep}}$).

All architectural and optimization settings remain as specified above. The key difference from SemlaFlow is the use of a coordinate-space, shell-bound probability path rather than a latent-space Gaussian path.

K.6 Inference and Evaluation

Sampling procedures. All models use their respective backbone's default sampling procedure:

- **GeoDiff/SubGDiff:** Ancestral sampling (DDPM) with $T = 5000$ (QM9) or $T = 500$ (Drugs) steps.
- **EDM/SSD-EDM:** Probability-flow ODE with 1000 function evaluations using the Heun solver.
- **SemlaFlow/SSD-Flow:** Deterministic ODE integration with 100 NFEs using Dopri5.
- **MCF/SSD-MCF:** DDPM sampling with 1000 steps (default) or DDIM with 50 steps (fast sampling ablation).

For SSD variants, the shell initialization is applied at inference time:

$$\mathbf{x}_T^{(i)} = r_{\text{SSD}} \frac{\mathbf{v}^{(i)}}{\|\mathbf{v}^{(i)}\|}, \quad \mathbf{v}^{(i)} \sim \mathcal{N}(0, I), \tag{53}$$

with a fresh random permutation π for atom-shell assignments.

Evaluation protocols. We follow the exact evaluation protocols defined by each backbone:

- **Conditional generation (GeoDiff/SubGDiff):** SubGDiff protocol with RMSD thresholds 0.5 Å (QM9) and 1.25 Å (Drugs), using the official evaluation script from `https://github.com/IDEA-XL/SubgDiff`.
- **Conditional generation (MCF):** Original MCF protocol with RMSD threshold 0.5 Å, using the official implementation from `https://github.com/apple/ml-mcf`.
- **Unconditional refinement (EDM/SemlaFlow):** SemlaFlow protocol with stability checks (1 Å neighborhood), validity (RDKit sanitization), and uniqueness (duplicate removal), using the official evaluation pipeline from `https://github.com/rssrwn/semla-flow`.

All metrics are computed using the same preprocessing, alignment procedures, and post-processing steps as the original papers to ensure fair comparison.

K.7 Computational Environment

Hardware. All experiments were conducted on NVIDIA A100 40GB GPUs. Training times per epoch are reported in Table 18 (Appendix M).

Software.

- **Framework:** PyTorch 2.1.0 with CUDA 12.1
- **Mixed precision:** Automatic mixed precision (AMP) training enabled
- **Molecular libraries:** RDKit 2023.03.1, OpenBabel 3.1.1
- **Numerical libraries:** NumPy 1.24.3, SciPy 1.10.1
- **Graph libraries:** PyTorch Geometric 2.3.0, DGL 1.1.0

Reproducibility. We use fixed random seeds (42 for training, 2023 for evaluation) across all experiments; Table 4 additionally uses training seeds 43 and 44. All code, trained checkpoints, and evaluation scripts are available at `https://github.com/ShellDiffusion/SSD-ShellDiffusion`.

K.8 Summary

In summary, SSD introduces minimal changes to existing backbones:

1. **Two auto-calibrated constants:** r_{SSD} and $d_{\min}$ (computed once per dataset).
2. **Two inherited schedules:** α_t and σ_t (mapped directly from backbone schedules).
3. **Three analytic drift terms:** Radial attraction, repulsion, and score correction (closed-form, no learnable parameters).

This design ensures that SSD acts as a true plug-in enhancement: it improves generation quality and stability without increasing model capacity, training cost, or inference time (Table 18).

L Additional Ablations on Shell Geometry, Dynamics, and Noise

This appendix provides extended ablation studies complementing those in the main text. We evaluate (i) the choice of shell radius for initialization, (ii) the role of geometric priors in the reverse process, (iii) sensitivity to the thin-shell noise scale σ_t and whether Gaussian baselines can replicate SSD by reducing their own noise, (iv) atom–shell assignment strategies, (v) radius calibration granularity (per-dataset vs. per-size-bin vs. per-molecule), (vi) stratified evaluation across molecular scales, (vii) calibrated radius vs. test-set mean, and (viii) behavior under distribution shift.

L.1 Shell Radius

We evaluate three radius families: *small shells* (0.01–5 Å), the dataset-adaptive *mean shell* based on the mean radius of centered training conformations (6.37 Å, including a 7 Å variant), and a *large shell* (10 Å). Table 10 shows that excessively small radii induce atomic collapse and degrade coverage, whereas overly large radii reduce sample quality. The mean shell provides the best overall performance and remains competitive across backbones.

Table 10: Ablation on shell radius for SSD-SubGDiff (QM9). COV-R (higher is better) and MAT-R (lower is better).

Initialization	**COV-R** ↑	**MAT-R** ↓
SubGDiff (Gaussian init)	90.9	0.246
Mean shell (no SSD dynamics)	91.0	0.245
Small shell (0.01 Å)	75.2	0.352
Small shell (1 Å)	82.0	0.310
Small shell (5 Å)	85.5	0.288
Mean shell (6.37 Å)	**93.2**	**0.238**
Mean shell (7 Å)	91.5	0.241
Large shell (10 Å)	84.0	0.290

L.2 Geometric Priors in the Reverse Dynamics

With the mean shell fixed, Table 11 evaluates the contribution of minimum-separation repulsion $d_{\min}$, thin-shell noise σ_t, and their combination. Repulsion stabilizes close-contact regions and prevents collapse, while thin-shell noise improves coverage by diversifying radial configurations. The combination yields the strongest performance, matching the full SSD dynamics.

Table 11: Ablation on geometric priors for SSD-SubGDiff (QM9). Mean shell (6.37 Å) used for all variants.

Variant	**COV-R** ↑	**MAT-R** ↓
SubGDiff (Gaussian init)	90.9	0.246
Mean shell (no SSD dynamics)	91.0	0.245
Mean shell (no priors)	91.0	0.245
Mean shell + $d_{\min}$	93.1	0.240
Mean shell + σ_t	92.7	0.240
Mean shell + $d_{\min}$ + σ_t (full SSD)	**93.2**	**0.238**

L.3 Shell Geometry vs. Noise Tuning

We address two questions: (i) Does SSD require careful tuning of the thin-shell noise σ_t? (ii) Can Gaussian baselines replicate SSD by reducing their noise scale? Table 12 shows that SSD remains stable when σ_t is scaled across five orders of magnitude, confirming that SSD does not rely on precise noise calibration. Table 13 shows that reducing Gaussian noise consistently degrades performance, because noise reduction cannot alter the underlying high-dimensional concentration at radius $\sigma_T\sqrt{3n}$. Together, these results confirm that SSD's advantages stem from its chemically scaled shell geometry, not from noise tuning.

Table 12: Robustness to thin-shell noise scaling for SSD-SubGDiff on GEOM-QM9. SSD uses a chemical shell at $r = 6.37$ Å. The baseline σ_t is inherited from SubGDiff without modification.

Variant	**COV-R** ↑	**MAT-R** ↓
$\sigma_t = 0$ (deterministic)	93.1	0.240
Baseline (inherited)	**93.2**	**0.238**
Baseline $\times 10$	93.0	0.239
Baseline $\times 10^2$	92.9	0.239
Baseline $\times 10^3$	92.8	0.240
Baseline $\times 10^4$	92.7	0.241
Baseline $\times 10^5$	92.5	0.242

Table 13: Reducing Gaussian noise does not replicate SSD on GEOM-QM9. Gaussian uses a dimensional shell at radius $\sigma_T\sqrt{3n}$ where σ_T is the forward noise scale and n is the number of atoms.

Variant	**COV-R** ↑	**MAT-R** ↓
Baseline (SubGDiff)	90.9	0.2460
$0.9\times$ baseline	85.3	0.2450
$0.8\times$ baseline	78.1	0.2720
$0.7\times$ baseline	75.8	0.3205
$0.5\times$ baseline	72.4	0.3520
$0.1\times$ baseline	60.3	0.3755
SSD (ours)	**93.2**	**0.2380**

L.4 Atom–Shell Assignment Strategies

SSD assigns shell points via a random permutation π, preserving permutation symmetry. Table 14 compares this to deterministic nearest-distance matching (NDM). NDM enforces fixed atom–shell correspondences, a condition never present during inference, which distorts the forward geometry and causes trajectories to deviate from the shell. As a result, NDM collapses coverage and severely increases MAT-R, highlighting the importance of random assignment.

Table 14: Ablation on atom–shell assignment (GEOM-QM9, SSD-SubGDiff). Random assignment is the default; NDM significantly degrades performance.

Assignment Strategy	**COV-R** $\uparrow$	**MAT-R** $\downarrow$
Random (default)	**93.2**	**0.2380**
Nearest-distance matching	0.0	1.7470

L.5 Radius Calibration: Per-Size-Bin and Per-Molecule Ablation

A natural concern is whether a single dataset-level shell radius r_{SSD} suffices for diverse datasets such as GEOM-Drugs, where molecules vary substantially in size. We address this with both a theoretical and empirical analysis.

Theoretical necessity of dataset-level radius. Diffusion requires a fixed, unconditional terminal distribution p_T. A per-molecule radius r_i would make p_T conditional on molecular identity, so the terminal distribution ceases to be well-defined as a marginal prior. In the unconditional refinement setting (EDM, SSD-Flow), molecular identity is unavailable at inference, so per-molecule calibration is fundamentally inapplicable.

Empirical analysis (conditional setting). In the conditional generation setting, where molecular identity is available, finer-grained calibration is in principle possible. We compare three strategies on GEOM-Drugs: (i) the dataset-level r_{SSD} used throughout the paper, (ii) per-size-bin radii (computed as the per-group training-set mean over size buckets 20–30, 30–50, ≥ 50 atoms), and (iii) a per-molecule radius r_i taken from each molecule's training conformation.

Table 15: Radius calibration ablation on GEOM-Drugs (conditional, SSD-SubGDiff). Size-bin radii are per-group training-set means. Even with full molecular identity available, finer calibration yields $< 0.3\%$ improvement over the dataset-level default.

Shell strategy	**COV-R** $\uparrow$	**MAT-R** $\downarrow$
Dataset-level $r_{\text{SSD}} = 21.16$ Å (default)	92.50	0.8300
Per-size-bin (20–30: 15.60; 30–50: 20.30; $\geq$50: 23.61 Å)	92.63	0.8282
Per-molecule r_i	92.78	0.8274

The reverse dynamics, driven by radial attraction toward the shell center combined with score-based local refinement, rapidly organize atoms into molecule-specific geometries regardless of the precise initialization radius. This explains why a single dataset-level radius is empirically sufficient: the shell sets the global scale, while molecule-specific structure is recovered by the learned score during the reverse process.

L.6 Stratified Evaluation Across Molecular Scales

A second concern is whether SSD's improvements degrade as molecules grow larger or move further from the calibration center. We address this with a stratified evaluation on the GEOM-Drugs test set, partitioning molecules by heavy-atom count.

The pattern is consistent with the scaling argument of Appendix B: the per-atom mismatch between a Gaussian prior and the data does not depend on n, but the absolute transport it induces grows as $\sqrt{n}$, so larger molecules require larger corrections—and SubGDiff degrades sharply (88.0 $\rightarrow$ 85.1 $\rightarrow$ 59.0

Table 16: Stratified evaluation on GEOM-Drugs by atom count (conditional, SubGDiff vs. SSD-SubGDiff). Weighted averages are consistent with the GEOM-Drugs row of Table 1 (SubGDiff ≈ 76.2, SSD-SubGDiff ≈ 92.5 COV-R). SSD's gain grows monotonically with molecule size, exactly where the Gaussian baseline degrades most.

Atom count	**# Molecules**	**SubGDiff COV-R**	**SSD-SubGDiff COV-R**	Δ
20–30	16	88.0	93.5	$+5.5$
30–50	641	85.1	93.0	$+7.9$
≥ 50	343	59.0	**91.5**	$\mathbf{+32.5}$

COV-R as molecules grow). SSD-SubGDiff, in contrast, remains stable across all size groups (93.5 $\rightarrow$ 93.0 $\rightarrow$ 91.5 COV-R). This stratification is observational: molecule size is not varied independently of other molecular properties. The gain Δ therefore widens monotonically with molecule size, reaching $+32.5$ COV-R for molecules with ≥ 50 atoms.

L.7 Calibrated Radius vs. Test-Set Mean

We further quantify how closely the auto-calibrated r_{SSD} (computed from the training set as described in Appendix K) matches the actual radial distribution of the test set. Using the identical computation on test conformations, we obtain:

Table 17: Calibrated r_{SSD} vs. test-set mean radius, computed identically.

Dataset	**Test-set mean (Å)**	r_{SSD} **(Å)**	**Deviation (Å)**
QM9	7.88	6.37	1.51
Drugs	22.84	21.16	1.68

Two observations: (i) The absolute deviation (1.51 Å on QM9, 1.68 Å on Drugs) is small relative to the shell radius itself, and well inside the regime where SSD remains robust—Table 10 shows that on QM9, radii from 5–7 Å all yield strong COV-R (85.5–93.2), with degradation only at extremes (0.01 or 10 Å). (ii) Even when r_{SSD} does not exactly match the test-set mean, the shell-aware reverse dynamics (radial attraction, repulsion, and score correction) actively compensate for moderate mismatch during sampling.

Robustness regime. This robustness is bounded but real. Table 10 establishes that on QM9, radii in the 5–7 Å range all yield strong COV-R (85.5–93.2), with degradation appearing only at extremes (0.01 or 10 Å). The $\sim$20% deviation between r_{SSD} and the test-set mean falls comfortably inside this operating range, and the shell-aware reverse dynamics actively compensate for the residual offset during sampling. SSD therefore requires r_{SSD} to lie within a broad chemically meaningful range, which all reasonable training-set statistics satisfy, without requiring fine-grained calibration to test-set means. This is consistent with the design philosophy in Section 2: SSD's role is to anchor the prior to molecular geometry rather than to a dimensional artifact, and a small additive offset within that geometric range does not undo the alignment.

L.8 Scope With Respect to Distribution Shift

Finally, we discuss how SSD behaves under cross-dataset transfer. Generative models depend on training data via two distinct mechanisms: (i) learned neural network weights θ, and (ii) dataset-dependent statistics. SSD introduces *exactly one* dataset-dependent quantity beyond what every coordinate-space generative model already needs: the scalar r_{SSD} (auto-calibrated as described in Appendix K), which is a one-pass training-set statistic computed in seconds, not a learned parameter. Applying SSD to a new dataset therefore requires only recomputing r_{SSD}; *no retraining is needed for the shell prior itself.*

What is and is not shown. SSD's shell prior can be recalibrated for a new dataset by recomputing r_{SSD}, whereas a Gaussian prior's scale is fixed by the noise schedule. This is a property of the prior's

construction, not evidence of robustness under distribution shift. Our evaluations calibrate and train SSD separately on GEOM-QM9 and GEOM-Drugs and establish within-distribution performance on each; we do not evaluate transfer when the calibration distribution differs from the test distribution.

Related evidence in the paper. (i) The same calibration procedure, applied independently to QM9 and Drugs ($r_{\text{SSD}} = 6.37$ vs. 21.16 Å), yields gains on both. (ii) Section 4.5 and Appendix L.6 show that within each dataset, SSD's gain over the Gaussian baseline grows with molecule size and drift length. (iii) Table 10 shows that performance depends on the radius: SSD exceeds the Gaussian baseline only near the calibrated value. Transfer to chemistries not represented in training (e.g., peptides, ionic species, metal-containing molecules, disconnected complexes) is untested.

L.9 Summary

Across all ablations, we observe four consistent principles: (1) radii must match molecular scale; (2) geometric priors such as $d_{\min}$ and σ_t are beneficial only when paired with a shell-aligned forward process; (3) thin-shell noise acts as minimal stochastic support for the shell geometry, not as a tunable variance parameter; and (4) SSD's gains cannot be reproduced by reducing Gaussian noise or using deterministic atom–shell assignments. Together, these results show that SSD's improvements stem from a geometry-aligned forward–reverse process built on a chemically calibrated spherical shell. Only SSD aligns initialization, noise structure, and reverse dynamics into a single geometry-consistent framework.

M Efficiency Analysis

SSD introduces no additional parameters for coordinate-based diffusion backbones such as SubGDiff, and its training corruption is a closed-form interpolation with no solver or trajectory cache. SSD-Flow replaces SemlaFlow's encoder with a more compact EDM-style coordinate encoder and trains on the SSD path, keeping the number of function evaluations; it is therefore a different model from SemlaFlow rather than SemlaFlow with a new prior. In all SSD variants, batch sizes, optimizers, and sampling schedules remain identical to their corresponding baselines. The additional SSD drift terms reuse pairwise distances already computed by the backbone and thus introduce negligible overhead.

Training-time and parameter comparison. Table 18 summarizes parameter counts and per-epoch training time. SSD-SubGDiff matches SubGDiff exactly in both parameters and runtime, confirming that SSD imposes no extra computational burden on coordinate diffusions. In contrast, SSD-Flow achieves a substantially smaller encoder and faster training compared to SemlaFlow, owing to the use of a more compact EDM-style encoder.

Table 18: Parameter and runtime comparison for coordinate and latent-flow backbones on GEOM-QM9. SSD preserves coordinate-model runtime and reduces latent-flow cost.

Method	Encoder Type	Params (M)	Training Time/epoch (min)	Inference Time/step (ms)
SubGDiff	SubGDiff encoder	0.79	2.16	0.38
SSD-SubGDiff	SubGDiff encoder	**0.79**	**2.16**	**0.38**
SemlaFlow	SemlaFlow encoder	46.60	5.01	1.03
SSD-Flow	EDM-style encoder	**28.54**	**3.79**	**0.84**

Architectural rationale for SSD-Flow. SSD-Flow utilizes a streamlined EDM-style encoder compared to SemlaFlow. This architectural modification is necessary to adapt the latent-based SemlaFlow framework to the coordinate-space SSD formulation: since SSD defines priors directly on physical radii (r_{SSD}), it requires a backbone that operates in coordinate space rather than latent space. While this change also contributes to efficiency (reducing parameters from 46.6M to 28.54M), the performance gains are primarily attributed to the shell-guided prior and dynamics, as evidenced by our ablation studies (Table 2, where shell initialization consistently outperforms Gaussian baselines under identical architectures) and by the controlled Gaussian-Flow comparison in Appendix M.1 below.

M.1 Isolating the shell prior: Gaussian-Flow baseline

A direct concern raised in evaluation is whether the gains of SSD-Flow over SemlaFlow stem from the SSD shell prior or from the architectural change (replacing SemlaFlow's latent encoder with an EDM-style coordinate encoder). Maintaining SemlaFlow's latent-space encoder while imposing the SSD prior is infeasible by construction: SSD defines initialization on physical Cartesian radii (r_{SSD} in Å), which is incompatible with operating in a learned latent space where "radius" has no chemical meaning.

To isolate the contribution of the shell prior under a controlled architecture, we instead introduce a **Gaussian-Flow** baseline: identical to SSD-Flow in every respect (EDM-style coordinate encoder, flow-matching objective, substructure features, network depth, NFEs, optimizer, batch size, training steps), except that the prior is the standard Gaussian $\mathcal{N}(0, \sigma_T^2 I)$ rather than the spherical shell. This yields a strict three-way comparison in which the only varied factor between Gaussian-Flow and SSD-Flow is the prior.

Table 19: Isolating the shell prior in coordinate-space flow matching on GEOM-QM9. SemlaFlow and SSD-Flow are reproduced from Table 1; **Gaussian-Flow** (new) shares the EDM-style coordinate encoder of SSD-Flow but uses a Gaussian prior. Under identical encoder, SSD-Flow outperforms Gaussian-Flow by $+2.4$ Validity, $+4.9$ Uniqueness, and $+1.8$ MolStab—gains attributable solely to the shell prior.

Model	Encoder	Prior	Validity ↑	Uniqueness ↑	MolStab ↑
SemlaFlow	SemlaFlow enc.	Gaussian (latent)	99.4	95.4	99.7
Gaussian-Flow *(new)*	EDM-style enc.	Gaussian (coord)	97.6	95.1	98.1
SSD-Flow	EDM-style enc.	**Shell (coord)**	**100.0**	**100.0**	**99.9**

Interpretation. The Gaussian-Flow baseline achieves 97.6 Validity and 95.1 Uniqueness— *below* SemlaFlow's 99.4 and 95.4, despite using a smaller encoder. This isolates two effects that were previously entangled in the SemlaFlow→SSD-Flow comparison: (i) Moving from SemlaFlow's latent encoder to a coordinate-space EDM-style encoder *slightly reduces* performance under a Gaussian prior (-1.8 Validity, -1.6 MolStab); the lighter coordinate encoder by itself does not explain SSD-Flow's gains. (ii) Adding the shell prior on top of the same EDM-style encoder *recovers and surpasses* SemlaFlow's performance ($+2.4$ Validity, $+4.9$ Uniqueness, $+1.8$ MolStab over Gaussian-Flow). The shell prior is therefore the operative cause of SSD-Flow's improvement, not the encoder change. Combined with Table 2, where SSD also improves coordinate-space diffusion backbones (SubGDiff, EDM) under *zero architectural changes*, this provides architecture-independent evidence for the shell prior's contribution across both diffusion and flow matching paradigms.

Why the encoder change is necessary, not a confound. We emphasize that the encoder substitution in SSD-Flow is a structural requirement, not a design preference: SemlaFlow operates entirely in latent space, so a Cartesian-radius shell prior cannot be applied without first moving generation back to coordinate space. The Gaussian-Flow baseline demonstrates that, conditioned on this necessary architectural shift, the shell prior contributes the dominant share of the gain.

Inference cost. Inference cost remains unchanged: SSD uses the same number of SDE or ODE steps as each backbone, and the analytic radial and repulsion drifts contribute negligible cost relative to the SE(3)-equivariant layers that dominate runtime.

Summary. Across all backbones and datasets, SSD preserves computational cost for coordinate diffusions and reduces computation for latent-flow models, while substantially improving sampling stability and final accuracy. This confirms that SSD's benefits arise from its geometry-aligned formulation rather than increased model capacity or runtime.

N Scope and Design Choices

N.1 Scope of Applicability

SSD is designed as a **plug-in geometric enhancement for coordinate-space molecular generation models**. Our framework directly applies to any backbone that performs diffusion or flow matching on Cartesian coordinates $\mathbf{x} \in \mathbb{R}^{n\times 3}$, including:

- **Diffusion models**: GeoDiff, SubGDiff, EDM, GCDM, MuDiff
- **Flow matching**: Direct coordinate-space flows (e.g., SSD-Flow)
- **Field-based models**: MCF (which operates on coordinates despite field parameterization)

Latent-space models (e.g., GeoLDM [Xu et al., 2023], SemlaFlow [Irwin et al., 2025]): SSD can be applied by replacing the latent encoder with a coordinate-space encoder while preserving the downstream architecture. We demonstrate this with SSD-Flow (Section 4.1), which replaces SemlaFlow's latent encoder with an EDM-style coordinate encoder, achieving improved performance with 39% fewer parameters (Table 18).

Torsional-space models (e.g., Torsional Diffusion [Jing et al., 2022]): These methods perform diffusion on dihedral angles $\theta \in [0, 2\pi)^k$ rather than Cartesian coordinates. Extending SSD to torsional representations would require reformulating shell-based priors on the manifold $\mathrm{SO}(2)^k$, which is beyond the scope of this work but represents a natural direction for future research.

N.2 Design Philosophy

We deliberately prioritize **simplicity and generality** over task-specific optimizations:

Dataset-level calibration. SSD uses a single radius r_{SSD} (mean of training conformations) and minimum separation $d_{\min}$ per dataset. This requires no per-molecule tuning and generalizes robustly across molecular sizes (Section 4.5, Appendix L). Crucially, r_{SSD} is a *fixed dataset-level constant* rather than a time-varying or per-molecule quantity: it appears as a constant in the forward and reverse SDEs (Section 3), so the terminal distribution p_T remains a well-defined unconditional marginal prior, consistent with standard diffusion theory. Empirical evidence supporting this design—including a comparison of dataset-level, per-size-bin, and per-molecule calibration that shows $< 0.3\%$ improvement from finer-grained strategies—is provided in Appendix L.5.

No energy coupling. SSD provides geometric priors without explicit energy guidance, ensuring compatibility with any backbone. Energy-based refinement can be added orthogonally via post-processing or as an additional score term ($\mathbf{v}_{\text{score}} + \lambda\nabla E$).

Unimodal shell prior. Our spherical shell targets single-radius initialization, consistent with the unimodal-radius regime of GEOM-QM9 and GEOM-Drugs. Distributions with strongly multimodal radial structure—e.g., a dataset containing both small ligands and very large macromolecules whose radii do not overlap—would benefit from a mixture-of-shells prior, which still defines a fixed unconditional p_T and remains compatible with the diffusion formalism. We leave such extensions to future work.

N.3 What SSD Achieves

Key result. SSD improves quality (MAT-R/P) and diversity (COV-R/P) for all five backbones in paired comparisons under identical training budgets; the large gains are stable across three training seeds, and the smallest (MCF on GEOM-Drugs) is not statistically significant.

N.4 Limitations

We summarize the main limitations of SSD as a complement to the scope discussion above. (i) **Coordinate-space restriction.** SSD's shell prior is defined on Cartesian radii (r_{SSD} in Å) and is therefore directly applicable to coordinate-space backbones. Latent-space models (e.g., SemlaFlow [Irwin

Table 20: Summary of challenges addressed by SSD.

Challenge	Status
Gaussian concentration mismatch	✓ Solved (Table 2, Appendix B)
Long-range drift & spatial entropy	✓ Solved (Table 5, Section 4.5)
Computational overhead	✓ Solved (Table 18: zero added params)
Sampling efficiency	✓ Solved (Figure 2, row 2: robust to 20–100 steps)
SE(3)-equivariance	✓ Proven (Appendix I)
Forward-reverse consistency	✓ Proven (Appendix J: $O(\tau^2/r^2)$ error)

et al., 2025], GeoLDM [Xu et al., 2023]) require an architectural shift to a coordinate-space encoder before SSD can be applied (as we do in SSD-Flow); torsional-space models (e.g., Torsional Diffusion [Jing et al., 2022]) would require reformulating the shell prior on $\mathrm{SO}(2)^k$, which we leave to future work. (ii) **Single-radius shell prior.** The shell radius r_{SSD} is a fixed dataset-level constant—a structural requirement for p_T to be a valid unconditional marginal in the diffusion formalism (Appendix L.5). Datasets whose radial distribution is strongly multimodal would in principle benefit from a mixture-of-shells prior (which preserves a fixed unconditional p_T), but evaluating such extensions is outside the scope of this work; on the unimodal-radius distributions of GEOM-QM9 and GEOM-Drugs, finer-grained calibration yields negligible improvement (Table 15). (iii) **Statistical-significance reporting.** Unless marked otherwise, all tables report single runs (training seed 42). For representative pairs we report means and standard deviations over three training seeds (Table 4): the SubGDiff, EDM, and SSD-Flow gains are large relative to their spread, whereas the MCF gain on GEOM-Drugs is not statistically significant. GeoDiff and the ablations have not been repeated across seeds. (iv) **Benchmark coverage.** Evaluation focuses on the standard GEOM-QM9 and GEOM-Drugs benchmarks under each backbone's canonical protocol. Generalization to substantially different chemistries (e.g., metal-organic frameworks, peptides) is not evaluated and is governed primarily by the generalization of the underlying backbone weights, an axis orthogonal to the shell-prior design (see Appendix L.8 for discussion).

N.5 Broader Impact

SSD improves the accuracy and efficiency of 3D molecular geometry generation, with potential positive applications in drug discovery, materials science, and computational chemistry. By reducing sampling cost and improving stability for larger molecules, SSD can lower the compute barrier for academic and small-laboratory research groups working on conformer generation. SSD does not generate or optimize molecular properties (e.g., toxicity, binding affinity, synthesizability) and does not interface directly with experimental wet-lab or autonomous-discovery pipelines, so direct misuse pathways are limited. As a foundational improvement to coordinate-space generative models, the broader impacts of SSD mirror those of the underlying generative-modeling literature: improvements to the quality of molecular generators may be combined downstream with property-prediction or molecular-design pipelines, where standard responsible-research practices (target screening, dual-use review, and access controls for sensitive chemistries) apply at the application layer rather than at the geometry-generation layer addressed here. We are not aware of negative societal impacts that arise specifically from the SSD framework itself.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 claim (i) that the isotropic Gaussian prior's scale is set by the noise level and atom count rather than by chemistry, so that a per-atom scale mismatch induces an absolute transport discrepancy that grows with $\sqrt{n}$ (Appendix B, Section 4.5); (ii) a chemically scaled spherical-shell prior, whose radius r_{SSD} is a training-set statistic (Table 9), with forward radial attraction toward the shell and, for the score-based backbones, a geometry-guided hybrid reverse sampler in which short-range repulsion is applied only during sampling (Section 3); (iii) that shell initialization and shell-aware dynamics are needed jointly, supported by controlled ablations (Table 2); and (iv) paired improvements over five coordinate-space backbones under identical training and sampling budgets: SSD improves or ties every metric reported for both members of each pair (Table 1), and three-seed statistics for representative pairs confirm the SubGDiff, EDM, and SSD-Flow gains, while the MCF gain on GEOM-Drugs is sign-consistent but not statistically significant (Table 4, Section 4.1). The statement that SSD-augmented GeoDiff and EDM match or surpass stronger baselines is a cross-paper comparison against numbers reported in those papers, which the paper presents as secondary to the paired comparisons (Section 4.1, Appendix G). The plug-in claim is scoped in the paper as follows: SSD adds no learnable parameters and, for the score-based backbones, keeps the denoiser architecture, loss form, optimizer, and schedules, while the corruption distribution and regression target follow the shell-directed path (Appendix J.5, item C5); SSD-Flow additionally replaces SemlaFlow's encoder (Appendix M.1). Theoretical statements are limited to those listed under Question 3; the reverse process is not claimed to be the time reversal of the forward process, and its radial stability is supported empirically (Table 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix N.4 discusses (i) the coordinate-space restriction (latent-space backbones need a coordinate-space encoder, and torsional-space backbones a reformulated prior); (ii) the single dataset-level radius, which suits unimodal-radius data and is not designed for multimodal radius distributions; (iii) statistical reporting: three-seed statistics cover representative pairs only (Table 4), all other results are single runs, and the MCF gain on GEOM-Drugs is not statistically significant (see also Question 7); (iv) benchmark coverage: each dataset is calibrated and evaluated separately, so robustness is demonstrated within distribution only, and transfer to other chemistries (e.g., peptides, ionic or metal-containing species) is untested; (v) the theoretical status of the sampler: the reverse process is a geometry-guided hybrid sampler without a derived forward–reverse consistency guarantee; and (vi) SSD-Flow changes the encoder, an effect that the Gaussian-Flow control separates from that of the prior (Appendix M.1). Appendix N discusses scope and design choices further.

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper's theoretical statements and their assumptions are: (a) concentration of the Gaussian norm around $\sigma_T\sqrt{3n}$, a standard result stated with references in Appendix B; (b) SE(3)-equivariance of every SSD drift term, assuming an SE(3)-equivariant backbone score network, derived term by term in Appendix I; (c) linear, distance-independent contraction of the deterministic ($\sigma_t = 0$) forward path toward the assigned shell points, derived in Appendix J.1; and (d) for the score-based backbones, conditioned on the sampled shell target the implemented corruption kernel is exactly $\mathcal{N}(\bar{\mathbf{x}}_t, \tau_t^2\mathbf{I})$, so the regression target is its exact conditional score (Appendix J.5, item C5). We make no theoretical claim about the reverse process: it is a geometry-guided hybrid sampler rather than the derived time reversal of the forward process, no forward–reverse marginal-consistency or KL bound is claimed, and its radial stability is reported only as an empirical measurement (Appendix J.3, Table 5). Shell-thickness values (Appendix H) are measured or estimated quantities, not asymptotic bounds.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We build on the official implementations and evaluation protocols of all five backbones (GeoDiff, SubGDiff, EDM, SemlaFlow-style flow matching, MCF; Appendix K). For the score-based backbones, SSD keeps the denoiser architecture, loss form, optimizer, and schedules, and changes the corruption distribution and regression target; the closed-form corruption $\mathbf{x}_t = \bar{\mathbf{x}}_t + \tau_t \boldsymbol{\epsilon}$ is specified in Appendix J.5 (item C5) and Appendix K. SSD-Flow uses an EDM-style coordinate-space encoder (Appendix M.1). SSD adds two dataset-level quantities (Table 9): r_{SSD}, a one-pass training-set statistic, and $d_{\min} = 0.9$ Å, a fixed design bound below all covalent bond lengths in the data. Training configurations, optimizer settings, schedules, samplers, and seeds are given in Appendix K, and code and checkpoints are publicly available (Question 5).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code, training and evaluation scripts, and instructions are publicly available at `https://github.com/ShellDiffusion/SSD-ShellDiffusion` (linked in Section 1 and Appendix K); the repository links the processed data and trained checkpoints. All experiments use the public GEOM-QM9 and GEOM-Drugs datasets; preprocessing follows

the standard splits documented in Appendix E, identical to those used by GeoDiff, SubGDiff, EDM, and SemlaFlow.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 specifies datasets, evaluation protocols, and metrics. Appendix K provides full training configurations: optimizer (Adam), learning rates, batch sizes, training steps, schedules, and inference samplers for each backbone (Table 8). Optimization hyperparameters and schedules are inherited unchanged from each backbone; r_{SSD} is computed from the training set and $d_{\min}$ is fixed (Table 9). Appendix E details data splits, and Appendix D defines all evaluation metrics.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 4 reports means $\pm$ standard deviations (1σ across runs) over three training seeds (42/43/44) for representative paired comparisons: SubGDiff, EDM, and MCF on both datasets, and the matched Gaussian-Flow/SSD-Flow pair on GEOM-QM9. The variability captured is that of the full training run under fixed data splits and evaluation seed. The SubGDiff, EDM, and SSD-Flow gains are large relative to their spread; the MCF gain on GEOM-Drugs ($+0.76$, about 1.8σ of the paired difference) is reported as sign-consistent but not statistically significant. All other results, including Table 1, the GeoDiff pairs, and the ablations, are single runs (training seed 42), as stated in Appendix N.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on NVIDIA A100 40GB GPUs, as documented in Appendix K. Per-epoch training time and per-step inference time for each backbone are reported in Table 18 (Appendix M); for example, SSD-SubGDiff requires 2.16 minutes per epoch and SSD-Flow requires 3.79 minutes per epoch on a single A100. Combined with the per-backbone training schedules in Table 8, the total compute across all five backbones (training + ablations) is approximately 500–1000 GPU-hours on A100 40GB. Software environment (PyTorch 2.1.0, CUDA 12.1, RDKit 2023.03.1) is also listed.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics. Our work uses publicly available molecular benchmark datasets, involves no human subjects, and does not raise privacy, fairness, or dual-use concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impact in Appendix N.5. SSD improves the accuracy and efficiency of 3D molecular geometry generation, which has positive applications in drug discovery, materials science, and chemical research, and lowers the compute barrier for academic and small-laboratory research groups. The method does not generate or optimize molecular properties (e.g., toxicity, binding affinity, synthesizability) and does not interface with experimental wet-lab pipelines, so direct misuse pathways are limited. As a foundational generative-modeling improvement, the broader impacts mirror those typical of generative ML research, with standard responsible-research practices applying at the application layer rather than at the geometry-generation layer addressed here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: SSD is a geometric enhancement to molecular conformation generation models trained on standard public benchmarks (GEOM-QM9, GEOM-Drugs). It does not produce or optimize molecular properties and poses no high-risk misuse pathway analogous to large language models or image generators.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original papers for all five baseline backbones and for the datasets. The baseline implementations are used under their licenses: GeoDiff (`https://github.com/MinkaiXu/GeoDiff`, MIT), SubGDiff (`https://github.com/IDEA-XL/SubgDiff`, MIT), EDM (`https://github.com/ehoogeboom/e3_diffusion_for_molecules`, MIT), SemlaFlow (`https://github.com/rssrwn/semla-flow`, MIT), and MCF (`https://github.com/apple/ml-mcf`, Apple sample-code license distributed with the repository). The GEOM dataset, including the GEOM-QM9 and GEOM-Drugs subsets, is distributed under CC0 1.0 (Harvard Dataverse, doi:10.7910/DVN/JNGTDF).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the SSD code (training, evaluation, and SSD-specific modules) at `https://github.com/ShellDiffusion/SSD-ShellDiffusion`, together with links to the processed data and trained checkpoints. The repository includes README documentation, an environment specification, and commands for reproducing the main results. The code is released under the MIT License; files adapted from GeoDiff and EDM (MIT) and from MCF (Apple sample code license) keep their original licenses, which are listed with the affected files in the repository's `THIRD_PARTY_NOTICES.md`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or human subject research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a core component of the proposed methodology. Any use of LLMs in this work was limited to writing, editing, and formatting purposes, which per NeurIPS policy does not require declaration.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.