COFLO: FEDERATED SAMPLING OF MOLECULAR CONFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models for molecular conformer generation benefit from large, diverse training datasets. However, much of the available molecular data remains siloed across institutions with different chemical specializations, where privacy constraints and intellectual property concerns hinder centralized training. To address this challenge, we introduce CoFlo, a framework for collaborative conformer generation across distributed clients without sharing their training data or generative-model parameters. Clients independently train flow matching models on their local data, while a lightweight router is trained through federated learning. At inference time, the router combines the fixed client velocity fields to guide a shared sampling trajectory, using mixture weights motivated by a posterior-weighted decomposition of the joint velocity field. We evaluate CoFlo on GEOM-QM9 and GEOM-Drugs under IID and heterogeneous scaffold-, chemotype-, topology-based, and cross-dataset partitions. CoFlo outperforms average client performance across all four metrics in every evaluated setting and surpasses even the best individual client on all four metrics in most dataset–split combinations, including the realistic mixed-dataset setting. These results demonstrate the potential of inference-time collaboration of specialized experts for conformer generation.

1 INTRODUCTION

Molecular conformer generation is central to drug discovery and materials science. Given a molecular graph (atoms and bonds), the task is to produce an ensemble of physically valid, i.e., low-energy 3D geometries from which chemical properties can be calculated. Recent flow matching and diffusion models achieve strong results by learning to transform noise into molecular coordinates conditioned on graph topology (Frank et al., 2025; Xu et al., 2022; Jing et al., 2022). These models are typically trained on large, centralized conformer datasets. In practice, however, molecular data is inherently distributed. AI-driven approaches depend critically on data quality and quantity (Sarkar et al., 2024): insufficient training corpora, labeling biases, and batch effects between experimental sources all degrade model quality and generalization. Large-scale experimental datasets are often proprietary to commercial organizations, limiting accessibility for the broader research community (Wu et al., 2018). Non-IID data is a common challenge in FL for drug discovery (Guo et al., 2024). Different labs or pharmaceutical companies have datasets that focus on specific drug classes, occupy fundamentally different regions of chemical space (i.e., different scaffolds and chemotypes), and vary in size. Moreover, different datasets contain conformers generated at different levels of quantum-chemical theory. Intellectual property rights and competitive concerns further prevent direct data sharing (Sarkar et al., 2024; Oldenhof et al., 2024), while the cost of acquiring high-quality labeled conformer data makes centralizing diverse corpora practically infeasible. Federated learning (McMahan et al., 2017) offers a way around this: each client in the network trains a model on its own data and shares only model updates, not the molecular structures themselves. This paradigm is already proven at scale in drug discovery: the MELLODDY consortium applied FL across ten pharmaceutical companies to collaboratively train QSAR models on 2.6 billion experimental data points without any party sharing its proprietary compound library (Oldenhof et al., 2024). Beyond property prediction, FL has been applied to molecular generation via federated GANs (Zeng et al., 2021), to 3D diffusion models (Song et al., 2025), and to personalized generative frameworks that cluster clients by chemical similarity (Guo et al., 2024). Privacy-preserving knowledge distillation across institutions has also emerged as a complementary approach (Hanser et al., 2024), and dedicated surveys now document

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

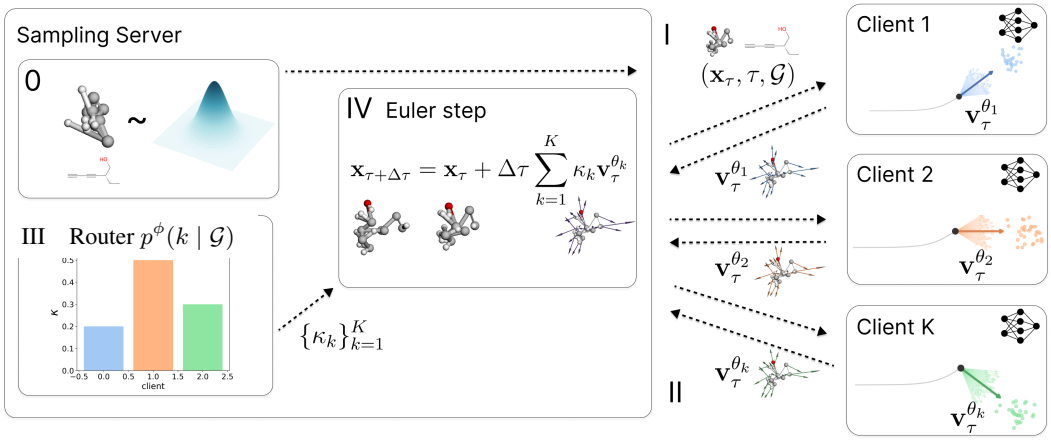


Figure 1: **CoFlo federated sampling.** The sampling server initializes coordinates from Gaussian noise (0) and broadcasts the graph, state, and time to clients (I). Clients return velocity predictions without sharing expert parameters (II). A graph-only router supplies fixed mixing weights (III), and the server takes a weighted Euler step (IV). Steps I, II, and IV repeat along the trajectory; the graph is sent only once and router weights can be cached.

the breadth of this landscape (Hanser, 2023; Savoia et al., 2026). This is a natural *federated learning* setting: each client holds a private conformer dataset, trains a local generative model, and we wish to leverage collective knowledge at inference without centralizing the data.

We propose *federated sampling*: clients train local flow matching models independently, and a lightweight router trained via federated learning, i.e., without sharing private data, combines their velocity fields at inference. Moreover, to benchmark our method, we extend the scaffold-based split from previous work by Zhu et al. (2022) with two more chemically heterogeneous synthetic splits, and add a cross-dataset split on GEOM, which simulates a real-life scenario. CoFlo outperforms average client performance across all four metrics in every evaluated setting and surpasses even the best individual client on all four metrics in most dataset–split combinations, including the realistic mixed-dataset setting.

2 Related Work

Molecular conformer generation. Distance geometry (Ganea et al., 2021), torsion-based (Jing et al., 2022), and coordinate-based (Xu et al., 2022) generative models predict 3D conformers from molecular graphs. Flow matching approaches (Lipman et al., 2023) learn continuous transport from noise to data, offering stable training and flexible architectures. Recent work has extended flow matching to conformer generation (Hassan et al., 2024) and accelerated sampling via SO(3)-averaged objectives and reflow (Cao et al., 2025). Our work builds on a DiT-based flow matching framework for conformers (Frank et al., 2025).

Federated and distributed learning for molecules. Federated learning (McMahan et al., 2017) has been applied to molecular property prediction and drug discovery. Zhu et al. (2022) introduced FedChem, the first federated benchmark for molecular property prediction, and introduced scaffold-based data partitioning as the dominant simulation of chemical heterogeneity in the molecular FL setting. We adopt their scaffold split and extend it with chemotype-based, topology-based, and cross-dataset partitions for the generative conformer setting. Xiong et al. (2020) study federated property prediction of drugs, while Hanser (2023) survey the broader federated drug-discovery landscape, distinguishing horizontal FL (partners in disjoint chemical spaces, the common case) from vertical FL. FedEDM (Song et al., 2025) is the closest prior work to ours: it federates an E(3)-equivariant diffusion model for 3D molecule generation, introducing parameter splitting and quantized updates to reduce communication cost. Our focus is inference-time combination of fixed pretrained experts under chemical heterogeneity. This avoids communication of generative-expert parameters; the lightweight router still exchanges parameter updates during training. Zeng et al.

(2021) combine federated learning with graph neural networks and generative adversarial networks for privacy-preserving 3D molecule generation (FL-DISCO). Guo et al. (2024) propose MolCFL, a personalized clustered federated framework that groups clients by molecular similarity to address non-IID data. A comprehensive survey of this landscape, spanning property prediction, virtual screening, and molecular design, is provided by Savoia et al. (2026).

Beyond federated learning, privacy-preserving collaboration for molecular data has been studied via secure multiparty computation (Hie et al., 2019) and data-driven knowledge distillation (Hanser et al., 2024). These approaches are complementary: our router framework is agnostic to the client training protocol and could in principle be combined with any of them.

Decentralized diffusion models. DDM (McAllister et al., 2025) showed that partitioning image data, training isolated experts, and routing via a learned classifier outperforms monolithic training. As opposed to DDM, we train a router in a federated manner without direct access to client structures, and employ a graph-only router operating on $\mathcal{G}$, removing any dependence on intermediate noisy coordinates $\mathbf{x}_\tau$ or diffusion time τ to maximize computational and communication efficiency. We evaluate the efficacy of our method on multiple heterogeneous splits, including established and novel ones.

3 Problem Setting

Decentralized conformer generation. K clients each hold a private dataset $\mathcal{D}_k = \{(\mathcal{G}_i, \mathbf{x}_{1,i})\}$ of molecular graphs with ground-truth conformers. The datasets are drawn from heterogeneous chemical spaces: clients may specialize in different molecular scaffolds, use different levels of conformer theory, or simply cover disjoint regions of drug-like chemical space. Each client independently trains a flow matching model f_{θ_k} on $\mathcal{D}_k$. At inference, given a new molecular graph condition $\mathcal{G}$, we wish to generate conformers $\mathbf{x}_1$ by combining the K pretrained models, without access to any $\mathcal{D}_k$ and without requiring parameter aggregation.

Flow matching background. For a fixed molecular graph $\mathcal{G}$, a conditional probability path $p_\tau(\mathbf{x} \mid \mathbf{x}_1, \mathcal{G})$ and its velocity $\mathbf{u}_\tau(\mathbf{x} \mid \mathbf{x}_1, \mathcal{G})$ induce the marginal field

$$\mathbf{v}_\tau(\mathbf{x} \mid \mathcal{G}) = \mathbb{E}\big[\mathbf{u}_\tau(\mathbf{x} \mid X_1, \mathcal{G}) \mid X_\tau = \mathbf{x}, \mathcal{G}\big]. \tag{1}$$

The conditional and marginal flow-matching objectives have equal parameter gradients under squared-error regression (Lipman et al., 2023):

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}\big[\|\mathbf{v}^\theta(X_\tau, \tau, \mathcal{G}) - \mathbf{u}_\tau(X_\tau \mid X_1, \mathcal{G})\|_2^2\big], \tag{2}$$

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}\big[\|\mathbf{v}^\theta(X_\tau, \tau, \mathcal{G}) - \mathbf{v}_\tau(X_\tau \mid \mathcal{G})\|_2^2\big], \tag{3}$$

where both expectations use the same time distribution, target distribution, and conditional path. The objectives differ by a constant independent of θ. Our experts use the clean-coordinate regression recipe of the DiT-APE implementation (Frank et al., 2025), where the perturbed training input is

$$\mathbf{x}_\tau = (1-\tau)\mathbf{x}_0 + \tau\mathbf{x}_1 + \sigma\mathbf{z}, \qquad \sigma \geq 0, \tag{4}$$

where $\mathbf{z}$ is Gaussian noise, independent of $\mathbf{x}_0$ and $\mathbf{x}_1$, projected onto the zero-center-of-mass subspace, and σ controls the training perturbation strength. Each expert model learns to predict the clean coordinates $\hat{\mathbf{x}}_1 = f_{\theta_k}(\mathbf{x}_\tau, \tau, \mathcal{G})$. At inference, each expert uses the velocity parameterization

$$\mathbf{v}^{\theta_k}(\mathbf{x}, \tau, \mathcal{G}) := \frac{f_{\theta_k}(\mathbf{x}, \tau, \mathcal{G}) - \mathbf{x}}{1-\tau}, \qquad 0 \leq \tau < 1, \tag{5}$$

to integrate samples from noise to data using the Euler method.

Key question. How should we combine K client velocity fields $\mathbf{v}^1, \ldots, \mathbf{v}^K$ into a single trajectory under chemical heterogeneity?

4 FEDERATED SAMPLING

For client distributions sharing a conditional probability path, the exact marginal field satisfies

$$\mathbf{v}_\tau(\mathbf{x} \mid \mathcal{G}) = \sum_{k=1}^{K} p_{\pi,\tau}(k \mid \mathbf{x}, \mathcal{G})\, \mathbf{v}_\tau^k(\mathbf{x} \mid \mathcal{G}), \tag{6}$$

where $\mathbf{v}_\tau^k$ is the corresponding client field and the weights are posterior client probabilities under a chosen client prior π (McAllister et al., 2025). Appendix F gives the derivation.

CoFlo defines the learned sampling field

$$\mathbf{u}_\tau^{\phi,\theta}(\mathbf{x}, \mathcal{G}) := \sum_{k=1}^{K} \underbrace{p^\phi(k \mid \mathcal{G})}_{\text{Router}} \underbrace{\mathbf{v}^{\theta_k}(\mathbf{x}, \tau, \mathcal{G})}_{\text{Client expert}}. \tag{7}$$

The graph-only weights approximate the generally state-dependent posterior in Eq. (6). Dropping the state dependence is exact only under the additional condition $k \perp\!\!\!\perp X_\tau \mid \mathcal{G}$, for example when client identity is determined by the graph on the modeled population. A partition of observed training graphs alone does not guarantee this condition for new molecules, IID partitions, or clients with different conformer distributions. We therefore evaluate graph-only routing as a modeling choice, rather than claiming exact recovery of the marginal field.

4.1 FEDERATED ROUTER TRAINING

In the standard DDM setup (McAllister et al., 2025), the router is trained on a centralized dataset with cluster labels. In our setting, each client k instead trains a local copy of the shared router using only its private molecular graphs, all labeled with its client identity k. No public or cross-client training examples are required. Both the router's graph encoder and classification head are trainable. The pretrained conformer-generation experts remain fixed.

Local training. At each communication round, clients initialize their local routers from the current global parameters and take a small number of gradient steps on the cross-entropy objective

$$\mathcal{L}_k(\phi) = \mathbb{E}_{(\mathcal{G},\mathbf{x}_1)\sim\mathcal{D}_k}\big[-\log p^\phi(k \mid \mathcal{G})\big]. \tag{8}$$

Although each local dataset contains only one client label, the shared router receives updates from all K classes over each communication round.

Aggregation. We average the client parameter updates with equal weights $1/K$, rather than weights proportional to local dataset size, so that each client contributes equally to the global update. The server applies FedAdam (Reddi et al., 2020) to this averaged update. This weighting targets the client-balanced objective $\mathcal{L}(\phi) = K^{-1}\sum_k \mathcal{L}_k(\phi)$, corresponding to an equal prior over clients, rather than weighting clients by their dataset sizes. Short local training intervals limit drift toward an individual client's class before updates from the other clients are incorporated. Our default uses five local gradient steps before parameter aggregation; the evaluated settings are detailed in Appendix A.2.

4.2 SAMPLING

In the proposed distributed protocol, the trained router is available to a sampling server, while generative-expert parameters remain on their clients. For a query graph $\mathcal{G}$, weights $\kappa_k = p^\phi(k \mid \mathcal{G})$ are constant along the trajectory and can be cached after one router evaluation. At each Euler step, the server sends the current state and time to the clients, receives their velocity predictions, and updates the shared state using Eq. (7); the graph needs to be transmitted only at setup. A client acting as sampling server evaluates its own expert locally. For experiments with $L > 1$ local integration steps, each expert advances its own copy of the shared state for L Euler steps before the server takes their router-weighted average. This agrees with per-step velocity mixing when $L = 1$, but generally differs for $L > 1$.

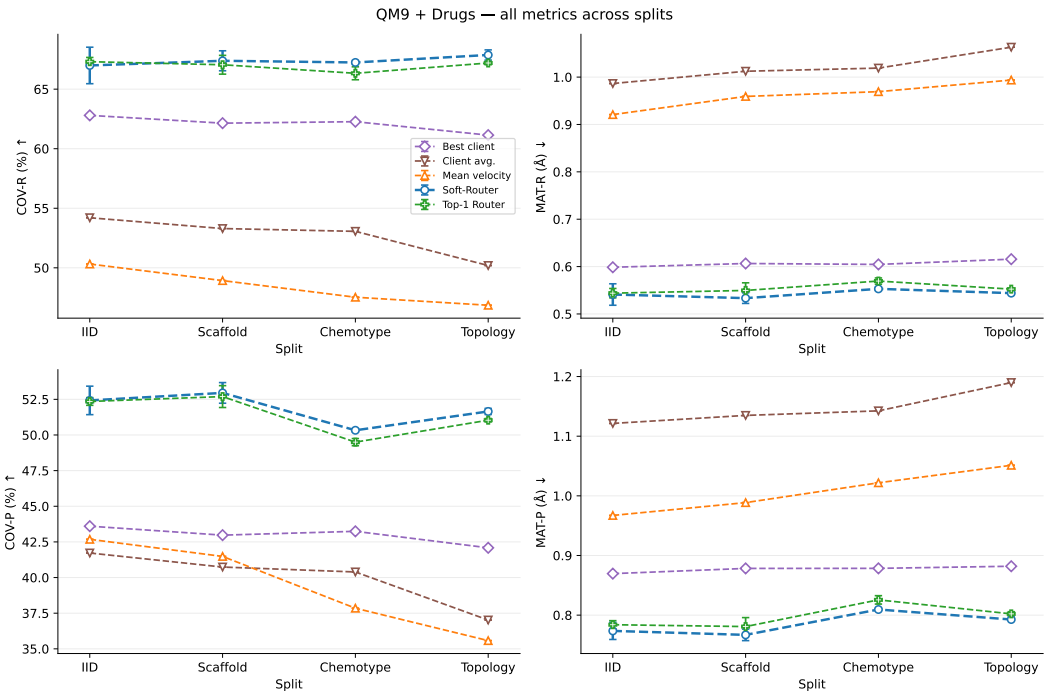


Figure 2: Conformer generation in the mixed QM9+Drugs federation. Error bars: seed standard deviation.

5 Experiments

5.1 Datasets

GEOM-QM9. The QM9 subset of GEOM (Axelrod & Gomez-Bombarelli, 2022) contains 133,258 small organic molecules with an average of 8.8 heavy atoms per molecule. Following Ganea et al. (2021), we use a split of 106,586 / 13,323 / 1,000 molecules for train / validation / test.

GEOM-Drugs. The Drugs subset of GEOM (Axelrod & Gomez-Bombarelli, 2022) contains approximately 304,000 drug-like molecules with an average of 24.9 heavy atoms per molecule. Its larger molecular graphs extend the evaluation beyond the small molecules in GEOM-QM9. Following Ganea et al. (2021), we use a split of 243,473 / 30,433 / 1,000 molecules for train / validation / test.

5.2 Non-IID Data Splits

Scaffold-based data partitioning was introduced by Zhu et al. (2022) for federated property prediction. We compare it with an IID control and two additional partitioning strategies that capture complementary dimensions of chemical heterogeneity. For each dataset, all four split regimes partition the training set into $K = 4$ disjoint client datasets.

(i) IID. Random assignment balanced across clients. This serves as a control with similar client data distributions; finite-sample and training variation can still produce differences between experts.

(ii) Scaffold. Molecules are grouped by their Bemis-Murcko scaffold and assigned to clients via greedy size balancing. Simulates labs specializing in different molecular scaffolds (Zhu et al., 2022).

(iii) Chemotype. Scaffolds are clustered by chemical similarity using Morgan fingerprints (radius 2, 2048 bits) with Tanimoto distance Tanimoto (1958) and Butina clustering Butina (1999). Simulates labs working on chemically related families spanning multiple scaffolds.

(iv) Topology. Same as chemotype, but atom types are replaced with wildcards before fingerprinting (radius 2), grouping molecules by graph structure regardless of element identity.

Mixed datasets. We combine four QM9 clients and four Drugs clients into an eight-client federation, using the same split regime within each dataset. This tests collaboration across distinct chemical domains.

We visualize the centroid and closest representative molecules of each client for the chemotype-based assignments in Appendix B.

5.3 METRICS

We adopt the four standard conformer-ensemble metrics introduced by Ganea et al. (2021). Given a generated set $\hat{S}$ and a reference set S, the measured metrics are:

Coverage (COV). Coverage measures the fraction of reference conformers that are well-approximated by the generated set:

$$\mathrm{COV}(\hat{S}, S) = \frac{1}{|S|} \left| \left\{ \mathbf{x} \in S \;\middle|\; \min_{\hat{\mathbf{x}} \in \hat{S}} \mathrm{RMSD}(\hat{\mathbf{x}}, \mathbf{x}) < \delta \right\} \right| \tag{9}$$

with threshold $\delta = 0.50\,\text{Å}$ for QM9 and $\delta = 0.75\,\text{Å}$ for Drugs. Mixed evaluations apply the corresponding threshold to each molecule. COV-R (recall) evaluates recall over reference conformers. COV-P (precision) swaps $\hat{S}$ and S to measure how many generated conformers are close to at least one reference.

Matching distance (MAT). Matching distance reports the mean nearest-neighbour RMSD from each reference conformer to the generated set:

$$\mathrm{MAT}(\hat{S}, S) = \frac{1}{|S|} \sum_{\mathbf{x} \in S} \min_{\hat{\mathbf{x}} \in \hat{S}} \mathrm{RMSD}(\hat{\mathbf{x}}, \mathbf{x}). \tag{10}$$

MAT-R (recall) uses reference-to-generated distances. MAT-P (precision) uses generated-to-reference distances. Lower MAT indicates better geometric fidelity.

We target 1,000 test molecules per dataset and generate twice the reference-conformer count per molecule. Metrics are averaged over the evaluated molecules. The mixed runs contain 1,000 Drugs and 996 QM9 molecules. Mixed scores are molecule-count-weighted means over that union, with each molecule scored at its dataset's COV threshold.

5.4 BASELINES

We compare against Best client, Client avg., and Mean velocity, together with centralized references trained on each dataset separately. Best client selects the best individual-client score independently for each metric and is therefore an oracle. Client avg. reports the mean of the K individual-client scores, while Mean velocity averages the client velocity fields uniformly during sampling. No centralized model trained on the mixed union is available.

Model architectures, training settings, and sampling details are provided in Appendix A.

5.5 MAIN RESULTS

Figure 2 compares conformer generation quality in the mixed QM9+Drugs federation across four split regimes. This setting represents collaboration between clients with different chemical domains. Table 3 in Appendix D reports results for all three dataset settings. The comparisons below refer to mean scores across three random router training seeds.

Gains over individual clients. Soft-Router outperforms Client avg. across all four metrics in all 12 dataset–split combinations and surpasses the per-metric Best client oracle on all four metrics in 9 of 12 combinations. These include all heterogeneous QM9 splits, Drugs IID and Scaffold, and every

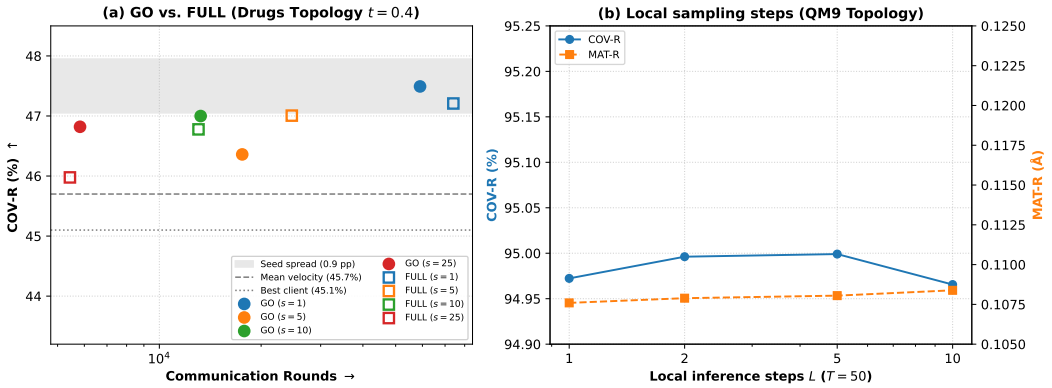


Figure 3: Router and sampling ablations: (a) GO/FULL routing with different local training steps on Drugs Topology. (b) QM9 Topology quality for local inference steps $L \in \{1, 2, 5, 10\}$, with $T = 50$ and $\delta = 0.50$ Å.

mixed-dataset split. The exceptions are QM9 IID, where CoFlo improves both MAT metrics but not COV, and Drugs Chemotype and Topology, where it improves COV-R but not the other three metrics. Thus collaborative inference improves on average client performance throughout the evaluation and exceeds even the strongest individual client in most settings.

Comparison with uniform aggregation. On QM9, Soft-Router improves mean COV-R over Mean velocity by 1.28, 0.88, and 2.47 percentage points on Scaffold, Chemotype, and Topology, respectively. The IID results are nearly identical (both 90.9% after rounding). On Drugs IID and Scaffold, Soft-Router COV-R is 45.96% and 46.15%, compared with 46.27% and 46.09% for Mean velocity. On Drugs Chemotype and Topology, routing improves COV-R over Mean velocity by 0.59 and 4.35 percentage points, respectively, and improves both MAT metrics and COV-P.

Mixed-dataset collaboration. CoFlo exceeds both Client avg. and Best client across all four metrics on every mixed-dataset split. Its COV-R gains over Best client range from 4.19 to 6.73 percentage points. Relative to Mean velocity, the COV-R gains are 16.67, 18.46, 19.71, and 21.01 percentage points on IID, Scaffold, Chemotype, and Topology, respectively. Even the mixed IID setting retains heterogeneity between QM9 and Drugs clients, despite random partitioning within each dataset. On mixed Chemotype and Topology, Soft-Router achieves the best mean scores across all four metrics among Best client, Client avg., Mean velocity, and Top-1 Router. Soft-Router also leads these methods on mixed Scaffold. On mixed IID, Top-1 Router has slightly higher COV-R (67.31% versus 66.99%), while Soft-Router performs better on the remaining metrics.

Efficient sampling. Top-1 Router closely matches Soft-Router, with COV-R differing by less than one percentage point on every mixed-dataset split. Its fixed expert selection permits sampling with a single client, reducing expert evaluations from KT to T if only the selected expert is executed. This saving is not measured by our current implementation, which evaluates all experts. Multiple local integration steps offer a complementary reduction in communication overhead, as discussed below.

5.6 Router Ablation

Figure 3(a) compares graph-only (GO) and noisy-state-conditioned (FULL) routers with $s \in \{1, 5, 10, 25\}$ local training steps. Both achieve their highest COV-R at $s = 1$, while GO and FULL differ by less than 0.85 percentage points at each setting, with no consistent benefit from state conditioning in this single-seed sweep. Figure 4 and Appendix H show relatively stable expert assignments along the sampling trajectory, suggesting limited additional benefit from conditioning routing on the noisy state.

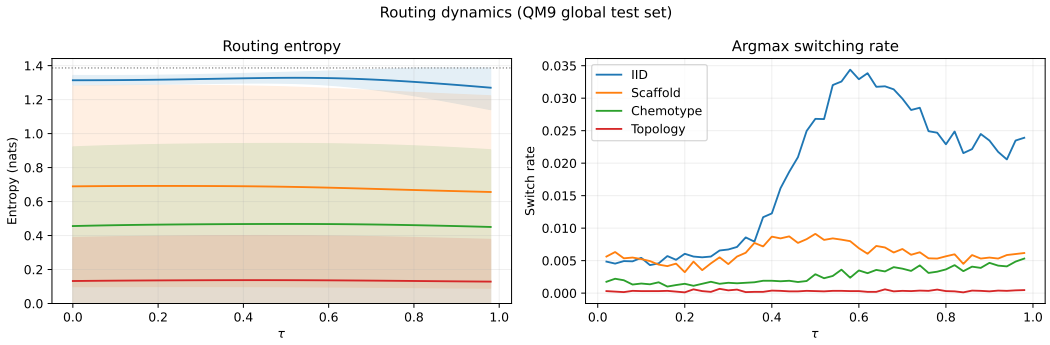


Figure 4: State-conditioned QM9 router diagnostic: routing dynamics over the ODE trajectory $\tau \in [0, 1]$ on the global test set. *Left*: routing entropy averaged over all conformers (lower = more decisive). Shading shows one standard deviation across conformers, clipped to the entropy range. *Right*: fraction of conformers whose argmax expert switches between consecutive steps (lower = more stable).

5.7 Varying local sampling steps

Taking L local integration steps before merging client states reduces communication rounds from T to $\lceil T/L \rceil$, while retaining T expert evaluations per client. This offers a direct way to reduce communication overhead, at the cost of allowing client trajectories to diverge between merges. Figure 3(b) shows $L \in \{1, 2, 5, 10\}$ on QM9 Topology with a fixed router, $T = 50$, and the same 996 molecules. Across this range, COV-R varies by only 0.034 percentage points and MAT-R rises from 0.1076 to 0.1084 Å (about 0.7%), while merge rounds fall from 50 to 5. Thus moderate local blocks largely preserve measured quality.

5.8 Routing Dynamics

Figure 4 examines how concentrated and stable router assignments are along the sampling trajectory. On Topology, routing entropy stays below 0.14 nats and fewer than 0.07% of conformers change their highest-weight expert between consecutive steps. Concentrated weights motivate the Top-1 approximation, while stable assignments support using a graph-only router rather than repeatedly adapting to the noisy state.

5.9 Sampling Time and Communication

Taking more local steps between merges reduces communication delays. We estimate the sampling time using the message sizes and model execution times in Table 1, measured for batches of 64 on an NVIDIA A100. Each expert performs $T = 50$ integration steps, with L local steps between merges.

Table 1: Message sizes and model execution times for a batch of 64. Timings use an NVIDIA A100.

Dataset	Graph s_c	State s_p	Expert t_{fwd}	Router t_{router}
QM9	418 kB	14.8 kB	2.1 ms	0.6 ms
GEOM-Drugs	1.30 MB	34.7 kB	4.1 ms	0.8 ms

Assuming clients compute and communicate in parallel, the time to generate one batch is approximated by

$$t_{\text{total}}(L) = \underbrace{\frac{t_{\text{rtt}}}{2} + \frac{8s_c}{B}}_{\text{one-time setup}} + \underbrace{T t_{\text{fwd}}}_{\text{expert compute}} + \underbrace{\left\lceil \frac{T}{L} \right\rceil \left(t_{\text{rtt}} + \frac{16 s_p}{B} \right)}_{\text{communication between merges}} . \tag{11}$$

Here t_{rtt} is the round-trip time, B is bandwidth per client in bits/s, and message sizes are in bytes. The graph is sent once, followed by s_p bytes in each direction per merge. We assume router computation

Table 2: Time to generate a batch of 64 conformers (ms). Estimates from Eq. (11), rounded to whole milliseconds. All settings use 50 integration steps per expert. Increasing the number of local steps L reduces communication rounds while keeping expert computation unchanged.

Dataset	Deployment	$L = 1$	$L = 2$	$L = 5$	$L = 10$	$L = 25$	$L = 50$
	Communication rounds	50	25	10	5	2	1
QM9	Same-machine	111.6	108.5	106.6	106.0	105.6	105.5
	Same-datacenter	131.8	118.7	110.8	108.2	106.6	106.1
	Same-region	625.2	369.3	215.7	164.5	133.8	123.6
	Cross-continent	7831.8	4022.6	1737.1	975.3	518.2	365.8
GEOM-Drugs	Same-machine	213.9	210.0	207.6	206.9	206.4	206.2
	Same-datacenter	234.1	220.2	211.8	209.1	207.4	206.8
	Same-region	748.2	484.3	326.0	273.2	241.5	231.0
	Cross-continent	8161.6	4272.8	1939.5	1161.8	695.1	539.6

overlaps the first expert call and its weights are cached, although the evaluated sampler recomputes them at each merge. The estimate excludes server merging, network contention, protocol overhead, and differences in client speed.

Table 2 assumes 10 Gbps for same-machine and same-datacenter links, with round-trip times of 0.1 and 0.5 ms. Same-region links use 1 Gbps and 10 ms, and cross-continent links use 100 Mbps and 150 ms. Fast links keep sampling time close to the expert computation time of 105 ms for QM9 and 205 ms for Drugs. On slower links, less frequent merging matters much more. For Drugs across continents, increasing L from 1 to 25 reduces the estimate from 8.2 to 0.7 s, a roughly twelvefold speedup. Distributed runtimes were not measured. The effect of L on generation quality was evaluated only on QM9 (Section 5.7 and Appendix J).

Pipelining can increase throughput when generating many batches: while one batch waits for a client response, another can use available compute capacity. Each batch still follows the same sequence of integration and communication steps. The potential gain depends on available compute and network capacity and is not measured here or included in Table 2.

6 Limitations

Graph-only routing may be insufficient when clients have different conformer distributions for the same molecular graph. Keeping data local does not guarantee privacy: router outputs can reveal client association, and individual velocities may leak information. Secure aggregation could protect individual messages under an explicit threat model (Appendix I), but is not implemented here.

7 Conclusion

CoFlo enables collaborative conformer generation by combining independently trained client experts with a lightweight federated router. Across QM9, Drugs, and mixed federations under all four split regimes, Soft-Router improves all four metrics over Client avg. It also outperforms Mean velocity and the per-metric Best client on all four metrics in 10 and 9 of the 12 dataset–split combinations, respectively, with particularly strong gains in mixed-dataset collaboration which reflects a real life scenario. Our analyses identify complementary routes to better efficiency: Top-1 routing retains similar coverage while permitting single-expert sampling, and multiple local integration steps reduce communication rounds while keeping expert integration work fixed. Together, these findings establish the value of collaborative sampling for combining local chemical expertise without centralizing training data or retraining the generative experts.

Reproducibility Statement

Appendix A details the architectures, training, and sampling settings, and Appendix D provides full numerical results.

AI ASSISTANCE

We used generative AI for literature retrieval and discovery, research ideation and execution (including code development and results analysis), and drafting and polishing the manuscript. The authors take responsibility for the final text, claims, code, and results.

REFERENCES

Simon Axelrod and Rafael Gomez-Bombarelli. GEOM: Energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9:185, 2022. doi: 10.1038/s41597-022-01288-4.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

Darko Butina. Unsupervised data base clustering based on daylight's fingerprint and tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, 39(4):747–750, 1999.

Zhonglin Cao, Mario Geiger, Allan Dos Santos Costa, Danny Reidenbach, Karsten Kreis, Tomas Geffner, Franco Pellegrini, Guoqing Zhou, and Emine Kucukbenli. Efficient molecular conformer generation with so (3)-averaged flow matching and reflow. *arXiv preprint arXiv:2507.09785*, 2025.

J. Thorben Frank, Winfried Ripken, Gregor Lied, Klaus-Robert Müller, Oliver T. Unke, and Stefan Chmiela. Sampling 3D molecular conformers with diffusion transformers. *arXiv preprint arXiv:2506.15378*, 2025.

Octavian-Eugen Ganea, Lagnajit Pattanaik, Connor W. Coley, Regina Barzilay, Klavs F. Jensen, William H. Green, and Tommi Jaakkola. GeoMol: Torsional geometric generation of molecular 3D conformer ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Yan Guo, Yongqiang Gao, and Jiawei Song. Molcfl: A personalized and privacy-preserving drug discovery framework based on generative clustered federated learning. *Journal of biomedical informatics*, 157:104712, 2024.

Thierry Hanser. Federated learning for molecular discovery. *Current Opinion in Structural Biology*, 79:102545, 2023. doi: 10.1016/j.sbi.2023.102545.

Thierry Hanser et al. Data-driven federated learning in drug discovery with knowledge distillation. *Nature Machine Intelligence*, 2024. 7/432-436.

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.

Brian Hie, Hyunghoon Cho, and Bonnie Berger. Realizing private and practical pharmacological collaboration. *Science*, 362(6409):347–350, 2019. doi: 10.1126/science.aat4807.

Bowen Jing, Gabriele Corso, Jeffrey Chang, Regina Barzilay, and Tommi Jaakkola. Torsional diffusion for molecular conformer generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

David McAllister, Matthew Tancik, Jiaming Song, and Angjoo Kanazawa. Decentralized diffusion models. *arXiv preprint arXiv:2501.05450*, 2025.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54, pp. 1273–1282. PMLR, 2017.

Martijn Oldenhof et al. MELLODDY: Cross-pharma federated learning at unprecedented scale unlocks benefits in QSAR without compromising proprietary information. *Journal of Chemical Information and Modeling*, 64(7):2331–2344, 2024. doi: 10.1021/acs.jcim.3c00799.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Chandrani Sarkar et al. Current strategies to address data scarcity in artificial intelligence-based drug discovery: A comprehensive review. *Computers in Biology and Medicine*, 2024. doi: 10.1016/j.compbiomed.2024.108198.

Martina Savoia, Antonio Lavecchia, and Francesco Piccialli. Federated learning in drug discovery: Challenges, innovations, and future directions. In *Applied Artificial Intelligence for Drug Discovery*, pp. 557–581. Springer, 2026. doi: 10.1007/978-3-031-98022-0_18.

Zixing Song, Irwin King, and José Miguel Hernández-Lobato. FedEDM: Federated equivariant diffusion model for 3D molecule generation with enhanced communication efficiency. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion)*. ACM, 2025.

Taffee T Tanimoto. An elementary mathematical theory of classification and prediction. 1958.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530, 2018. doi: 10.1039/C7SC02664A.

Zhaoping Xiong et al. Facing small and biased data dilemma in drug discovery through federated learning. 2020.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. GeoDiff: A geometric diffusion model for molecular conformation generation. In *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022.

Zhenqi Zeng, Dong Cao, et al. FL-DISCO: Federated generative adversarial network for graph-based molecule drug discovery. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2021. doi: 10.1109/ICCAD51958.2021.9643440.

Wei Zhu, Jiebo Luo, and Andrew D. White. Federated learning of molecular properties with graph neural networks in a heterogeneous setting. *Patterns*, 3(6):100521, 2022. doi: 10.1016/j.patter.2022.100521.

A Implementation Details

A.1 Model Architectures

Client velocity field. Each client trains an independent flow matching model using the DiT-APE architecture proposed by Frank et al. (2025), a graph-conditioned Diffusion Transformer with absolute positional encoding. The model consists of 6 transformer layers with 8 attention heads and a hidden dimension of 256, conditioned via adaLN on the molecular graph topology through a MeshGraphNet (MGN) encoder with 2 layers and 256 features.

Router. The router is a 2-layer MeshGraphNet with 64 hidden features that processes solely the molecular graph $\mathcal{G}$, followed by a graph-level classification readout ($\texttt{segment_sum} \rightarrow \texttt{Dense}(64) \rightarrow \text{SiLU} \rightarrow \texttt{Dense}(K)$). It does not ingest 3D coordinates or diffusion timesteps. Its graph-only architecture avoids the coordinate-dependent transformer stack used by the generative experts.

A.2 Training Details

We use $\sigma = 0.5$ for the training perturbation in Eq. (4). The selected QM9 experts use 50 local training epochs and batch size 128. The selected Drugs experts use 10 local epochs and batch size 32. The mixed IID and Scaffold evaluations combine Drugs epoch-10 and QM9 epoch-50 checkpoints. Router federated training uses five local SGD steps per communication round, uniform client weighting, and a FedAdam server optimizer (Reddi et al., 2020). All main-table router settings report three seeds. Mixed Chemotype and Topology use Drugs experts at thresholds $t = 0.3$ and $t = 0.4$, respectively, together with QM9 experts at $t = 0.2$.

A.3 Sampling Details

CoFlo uses Soft-Router for the convex combination in Eq. (7). Top-1 Router instead assigns weight one to $k^* = \arg\max_k p^\phi(k \mid \mathcal{G})$ and zero to the other experts. For graph-only routing, this selection is fixed throughout the trajectory. Our implementation evaluates all expert fields before selecting the corresponding prediction; it does not measure the savings of querying only the selected expert.

Local integration-step sweep. The QM9 Topology sweep uses a fixed graph-only router without public data (seed 1, 25 local training steps) and $L \in \{1, 2, 5, 10, 25, 50\}$. Each expert performs $T = 50$ Euler steps, giving T/L merge rounds and KT expert evaluations per trajectory. All six runs contain the same 996 molecule IDs and 25,764 generated conformers; the $L = 1$ point belongs to this sweep. COV is read at $\delta = 0.50$ Each point is one run.

B Cluster Representatives

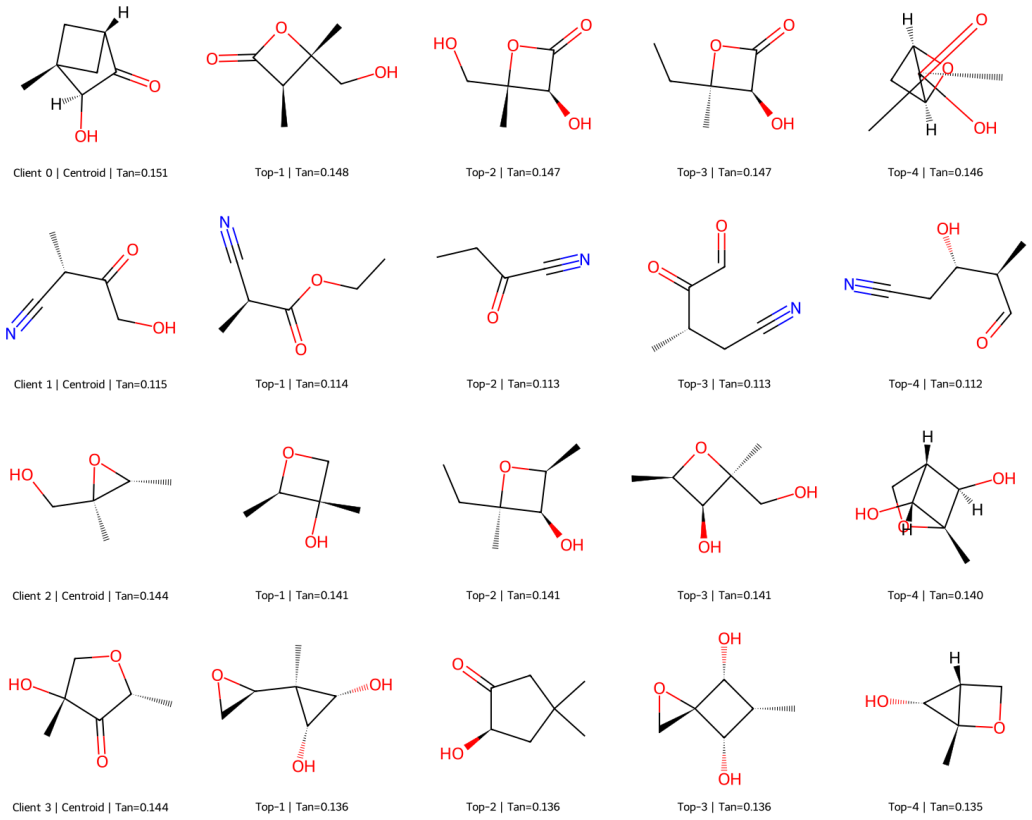


Figure 5: Centroid and four closest representative molecules for each of the four clients in the chemotype-clustered split (GEOM-QM9). Each row corresponds to one client. Column 1 shows the centroid (molecule with highest Tanimoto similarity to the mean Morgan fingerprint of the cluster); columns 2–5 show the four next-closest molecules. Tanimoto similarity to the cluster mean fingerprint is reported below each structure.

C Client Distributions across Federated Splits

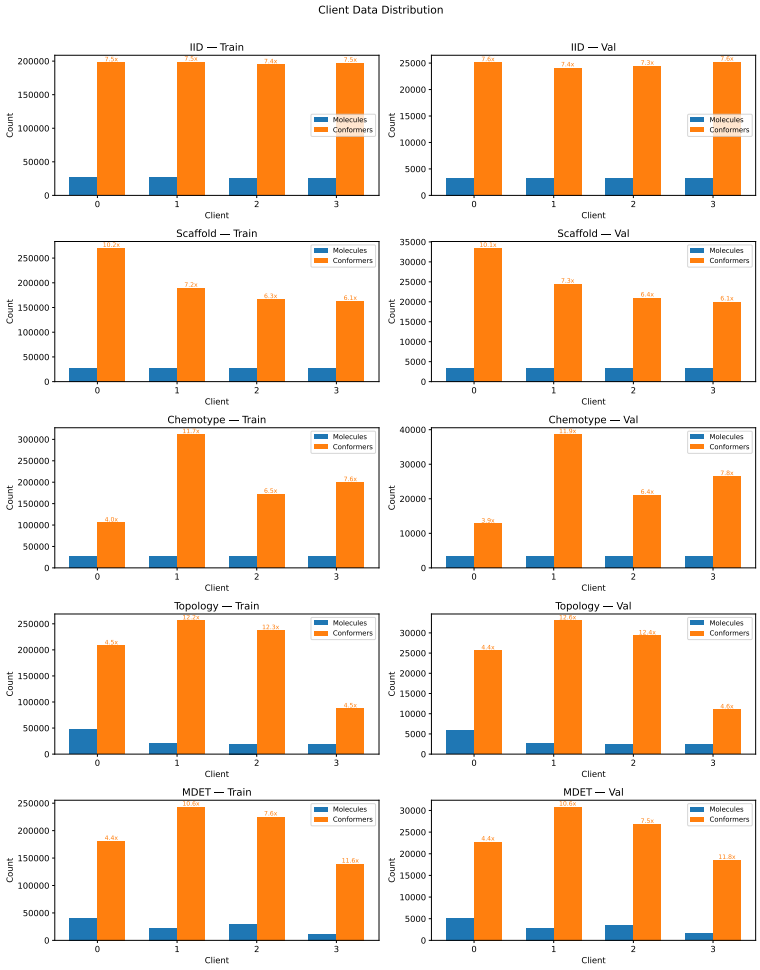


Figure 6: Molecule and conformer counts in the QM9 split diagnostics, shown separately for training and validation. MDET is an additional partition used only in supplementary diagnostics.

D Full Numeric Results

Table 3: Full results on GEOM-QM9, GEOM-Drugs, and their mixed federation ($K = 4$, 4, and 8). COV is reported in percent and MAT in Å. Soft-Router and Top-1 Router entries show mean $\pm$ sample standard deviation over three router seeds, except for Drugs Chemotype/Topology, which are single runs. Mixed scores average 996 QM9 and 1,000 Drugs molecules with dataset-specific thresholds. Best client is a per-metric oracle; centralized references use an unmatched configuration. Missing evaluations are marked by dashes. Training and checkpoint provenance are described in Appendix A.2.

		GEOM-QM9 ($\delta = 0.50$ Å)				**GEOM-Drugs** ($\delta = 0.75$ Å)				**QM9 + Drugs**			
Scenario	Method	COV-R ↑	MAT-R ↓	COV-P ↑	MAT-P ↓	COV-R ↑	MAT-R ↓	COV-P ↑	MAT-P ↓	COV-R ↑	MAT-R ↓	COV-P ↑	MAT-P ↓
IID	Centralized	94.3	0.102	89.8	0.125	79.3	0.457	72.2	0.570	–	–	–	–
IID	Best client	91.2	0.164	80.5	0.261	45.7	0.876	26.5	1.264	62.8	0.599	43.6	0.870
IID	Client avg.	90.8	0.168	80.1	0.265	45.2	0.884	26.2	1.269	54.2	0.986	41.7	1.121
IID	Mean velocity	90.9	0.160	80.1	0.259	46.3	0.871	26.4	1.262	50.3	0.921	42.7	0.967
IID	Soft-Router	90.9 ± 0.0	0.159 ± 0.000	80.1 ± 0.1	0.258 ± 0.001	46.0 ± 0.2	0.871 ± 0.003	26.6 ± 0.2	1.258 ± 0.002	67.0 ± 1.5	0.541 ± 0.023	52.4 ± 1.0	0.774 ± 0.014
IID	Top-1 Router	90.9 ± 0.2	0.167 ± 0.001	80.1 ± 0.3	0.264 ± 0.002	45.1 ± 0.2	0.885 ± 0.004	25.9 ± 0.1	1.269 ± 0.006	67.3 ± 0.4	0.544 ± 0.010	52.3 ± 0.3	0.784 ± 0.007
Scaffold	Centralized	94.3	0.102	89.8	0.125	79.3	0.457	72.2	0.570	–	–	–	–
Scaffold	Best client	90.6	0.179	79.8	0.279	45.2	0.887	26.3	1.266	62.1	0.607	43.0	0.878
Scaffold	Client avg.	89.8	0.186	78.6	0.290	45.0	0.890	26.1	1.269	53.3	1.012	40.7	1.135
Scaffold	Mean velocity	90.6	0.165	79.9	0.266	46.1	0.873	26.7	1.259	48.9	0.959	41.5	0.989
Scaffold	Soft-Router	91.9 ± 0.3	0.153 ± 0.002	81.8 ± 0.2	0.244 ± 0.001	46.2 ± 0.7	0.881 ± 0.002	26.8 ± 0.4	1.257 ± 0.002	67.4 ± 0.8	0.533 ± 0.011	53.0 ± 0.7	0.767 ± 0.010
Scaffold	Top-1 Router	92.2 ± 0.3	0.158 ± 0.002	81.6 ± 0.0	0.252 ± 0.001	45.2 ± 0.2	0.890 ± 0.004	26.2 ± 0.2	1.266 ± 0.003	67.0 ± 0.8	0.550 ± 0.016	52.7 ± 0.8	0.781 ± 0.015
Chemotype	Centralized	94.3	0.102	89.8	0.125	79.3	0.457	72.2	0.570	–	–	–	–
Chemotype	Best client	89.8	0.188	79.3	0.285	78.8	0.476	71.0	0.592	62.3	0.605	43.2	0.878
Chemotype	Client avg.	88.0	0.205	76.5	0.314	76.4	0.522	64.0	0.678	53.1	1.019	40.4	1.142
Chemotype	Mean velocity	90.6	0.170	79.7	0.274	78.4	0.500	65.6	0.661	47.5	0.969	37.9	1.022
Chemotype	Soft-Router	91.4 ± 0.2	0.157 ± 0.001	81.9 ± 0.2	0.245 ± 0.001	79.0	0.477	69.1	0.616	67.2 ± 0.2	0.553 ± 0.003	50.3 ± 0.1	0.809 ± 0.002
Chemotype	Top-1 Router	91.5 ± 0.1	0.160 ± 0.001	81.6 ± 0.0	0.251 ± 0.000	78.4	0.479	70.8	0.594	66.3 ± 0.5	0.569 ± 0.007	49.5 ± 0.3	0.826 ± 0.007
Topology	Centralized	94.3	0.102	89.8	0.125	79.3	0.457	72.2	0.570	–	–	–	–
Topology	Best client	86.5	0.210	75.5	0.313	79.6	0.455	72.6	0.566	61.1	0.616	42.1	0.882
Topology	Client avg.	79.0	0.300	65.4	0.435	65.5	0.629	49.5	0.821	50.2	1.063	37.0	1.190
Topology	Mean velocity	89.5	0.211	76.4	0.327	75.8	0.549	56.6	0.760	46.9	0.994	35.6	1.051
Topology	Soft-Router	92.0 ± 0.1	0.149 ± 0.000	82.9 ± 0.1	0.234 ± 0.001	80.2	0.461	71.0	0.592	67.9 ± 0.4	0.544 ± 0.001	51.7 ± 0.2	0.793 ± 0.001
Topology	Top-1 Router	92.0 ± 0.1	0.150 ± 0.001	83.0 ± 0.1	0.235 ± 0.001	79.7	0.457	72.2	0.570	67.2 ± 0.1	0.552 ± 0.002	51.0 ± 0.1	0.802 ± 0.004

E Score from Velocity Derivation

For a fixed graph $\mathcal{G}$, consider the idealized Gaussian path $X_\tau = \alpha_\tau X_1 + \beta_\tau X_0$, with $X_0 \sim \mathcal{N}(0, I)$ independent of X_1 conditional on $\mathcal{G}$. All coordinates, densities, and gradients can be expressed in an orthonormal basis of the zero-center-of-mass subspace. Let $\mathbf{v}_\tau$ be the marginal velocity induced by this path. Then

$$\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x} \mid \mathcal{G}) = -\beta_\tau^{-1} \mathbb{E}[X_0 \mid X_\tau = \mathbf{x}, \mathcal{G}], \tag{12}$$

$$\mathbf{v}_\tau(\mathbf{x} \mid \mathcal{G}) = \dot{\alpha}_\tau \mathbb{E}[X_1 \mid X_\tau = \mathbf{x}, \mathcal{G}] + \dot{\beta}_\tau \mathbb{E}[X_0 \mid X_\tau = \mathbf{x}, \mathcal{G}], \tag{13}$$

$$\mathbf{x} = \alpha_\tau \mathbb{E}[X_1 \mid X_\tau = \mathbf{x}, \mathcal{G}] + \beta_\tau \mathbb{E}[X_0 \mid X_\tau = \mathbf{x}, \mathcal{G}]. \tag{14}$$

Eliminating the conditional expectations gives

$$\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x} \mid \mathcal{G}) = \frac{\alpha_\tau \mathbf{v}_\tau(\mathbf{x} \mid \mathcal{G}) - \dot{\alpha}_\tau \mathbf{x}}{\beta_\tau(\dot{\alpha}_\tau \beta_\tau - \dot{\beta}_\tau \alpha_\tau)}. \tag{15}$$

This requires $\beta_\tau \neq 0$ and $\dot{\alpha}_\tau \beta_\tau - \dot{\beta}_\tau \alpha_\tau \neq 0$; endpoint limits require separate treatment. The identity does not directly establish a score conversion for the target-aligned, constant-noise training recipe in Eq. (4). Combining diffusion and flow experts would require compatible probability paths and time parameterizations, beyond merely sharing a Gaussian prior.

F Marginal Velocity Decomposition

Let π_k be a client prior and let $p_k(\mathcal{G}, \mathbf{x}_1)$ denote the client data distributions. The mixture is $p_\pi(\mathcal{G}, \mathbf{x}_1) = \sum_k \pi_k p_k(\mathcal{G}, \mathbf{x}_1)$. Assume a common conditional path and velocity $\mathbf{u}_\tau(\mathbf{x} \mid \mathbf{x}_1, \mathcal{G})$, independent of the client label once $(\mathbf{x}_1, \mathcal{G})$ are given. By the law of total expectation,

$$\mathbf{v}_\tau(\mathbf{x} \mid \mathcal{G}) = \mathbb{E}[\mathbf{u}_\tau(\mathbf{x} \mid X_1, \mathcal{G}) \mid X_\tau = \mathbf{x}, \mathcal{G}] \tag{16}$$

$$= \sum_k p_{\pi,\tau}(k \mid \mathbf{x}, \mathcal{G}) \mathbb{E}[\mathbf{u}_\tau(\mathbf{x} \mid X_1, \mathcal{G}) \mid X_\tau = \mathbf{x}, \mathcal{G}, k] \tag{17}$$

$$= \sum_k p_{\pi,\tau}(k \mid \mathbf{x}, \mathcal{G})\, \mathbf{v}_\tau^k(\mathbf{x} \mid \mathcal{G}). \tag{18}$$

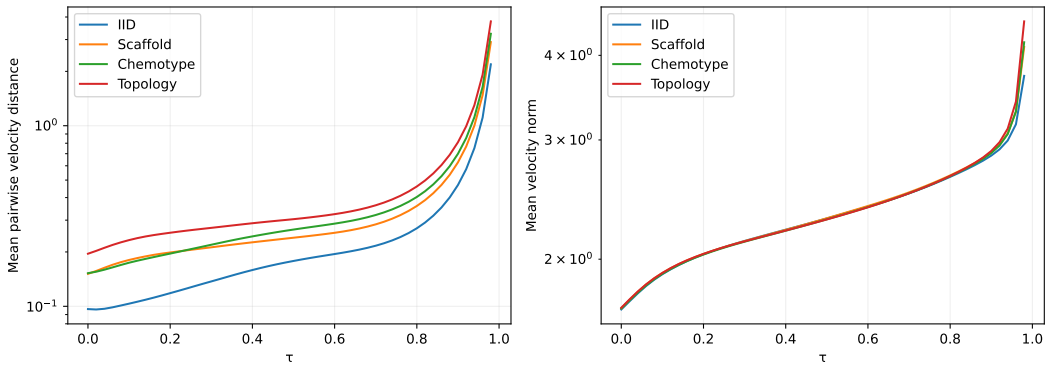


Figure 7: QM9 client velocity fields along the sampling trajectory: mean pairwise velocity distance (left) and mean velocity norm (right).

This proves Eq. (6) for the ideal client fields and chosen mixture prior. It does not require treating finite datasets as integration regions or assuming disjoint population supports.

Uniform client weighting in router training corresponds to $\pi_k = 1/K$ for the client-balanced data mixture. A dataset-size-weighted pooled mixture generally has a different prior. Moreover, $p_{\pi,\tau}(k \mid \mathbf{x}, \mathcal{G}) = p_\pi(k \mid \mathcal{G})$ requires $k \perp\!\!\!\perp X_\tau \mid \mathcal{G}$; this is not guaranteed by partitioning the observed training data. Replacing the posterior by the learned graph-only router and the ideal client fields by pretrained regressors defines Eq. (7) as an approximation. No exact mixture-transport guarantee for the implemented sampling rule is implied.

G Client Velocity Divergence

Figure 7 shows the lowest client-velocity disagreement on IID, as expected when clients are trained on samples from the same underlying distribution, and the highest disagreement on Topology, where clients represent structurally distinct molecular populations. Across all splits, velocity disagreement increases toward the end of the sampling trajectory. Differences between client coordinate predictions are amplified by the denominator $1 - \tau$ in the velocity parameterization, since at a shared state $\mathbf{v}^{\theta_i} - \mathbf{v}^{\theta_j} = (f_{\theta_i} - f_{\theta_j})/(1 - \tau)$, accentuating disagreement as $\tau \to 1$.

H Per-Molecule Routing Trajectories

Figures 8–12 show illustrative trajectories from QM9 router experiments with a different training recipe. They are not generated by the current graph-only router, whose weights are time-independent. The selected examples illustrate variation in expert concentration; they do not establish population-level switching rates or gains in conformer quality. MDET (also labeled MLFF in the analysis exports) appears only in supplementary diagnostics, not among the four main split regimes.

I Secure Aggregation of Client Velocity Fields

We outline a possible secure-aggregation extension following Bonawitz et al. (2016). It is not implemented in our experiments. At a given sampling step, each client knows its agreed routing weight κ_k and first forms its weighted velocity contribution.

Encoding and masked submission. Let Q encode real-valued vectors as fixed-point integers in $\mathbb{Z}_q^{3N}$, with a modulus and scale chosen to avoid overflow of the decoded aggregate. Client k encodes $\mathbf{a}_k = Q(\kappa_k \mathbf{v}^k)$. Each pair of participating clients establishes fresh pseudorandom masks satisfying $\mathbf{p}_{k,j} = -\mathbf{p}_{j,k} \pmod q$. Client k sends

$$\mathbf{y}_k = \mathbf{a}_k + \sum_{j \neq k} \mathbf{p}_{k,j} \pmod q. \tag{19}$$

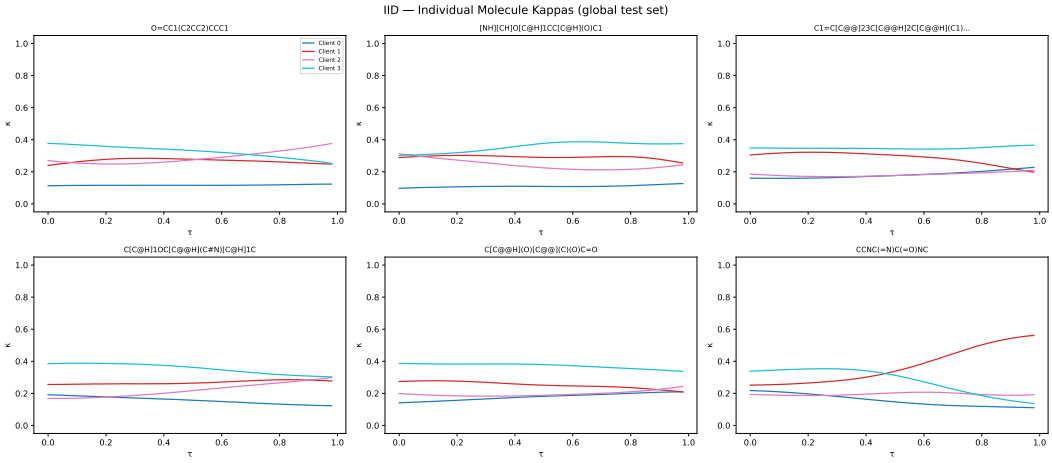


Figure 8: QM9 experiment: per-molecule $\kappa_k(\tau)$ trajectories for 6 randomly selected test molecules from **IID** split. The shown molecules have relatively diffuse expert weights.

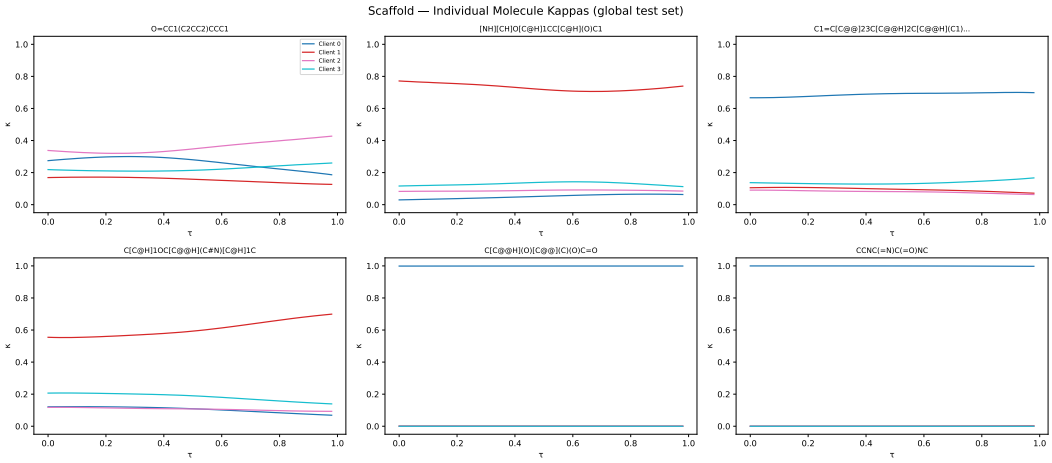


Figure 9: QM9 experiment: per-molecule $\kappa_k(\tau)$ trajectories from **Scaffold** split. Client trajectories mix on some molecules, for others one expert dominates.

The sampling server sums these messages:

$$\sum_k \mathbf{y}_k = \sum_k Q(\kappa_k \mathbf{v}^k) + \underbrace{\sum_k \sum_{j \neq k} \mathbf{p}_{k,j}}_{=0 \bmod q} = \sum_k Q(\kappa_k \mathbf{v}^k) \pmod q. \tag{20}$$

After decoding, this approximates the desired weighted field $\sum_k \kappa_k \mathbf{v}^k$ up to quantization error. Cancellation is exact in the encoded domain. Weighting must precede masking: an unweighted aggregate alone cannot recover an arbitrary weighted sum.

J Full Local Sampling-Step Sweep

Figure 13 and Table 4 retain all tested block sizes from the QM9 Topology sweep. At $L = 50$, each expert takes all 50 Euler steps before a single final weighted merge; the Euler step size and total expert work remain unchanged. Relative to $L = 1$, MAT-R increases by 12.2% and MAT-P by 20.9%, despite similar COV-R at $\delta = 0.50$ Å. Stable coverage therefore does not imply unchanged trajectories or geometric accuracy. The sweep settings and provenance are given in Appendix A.3.

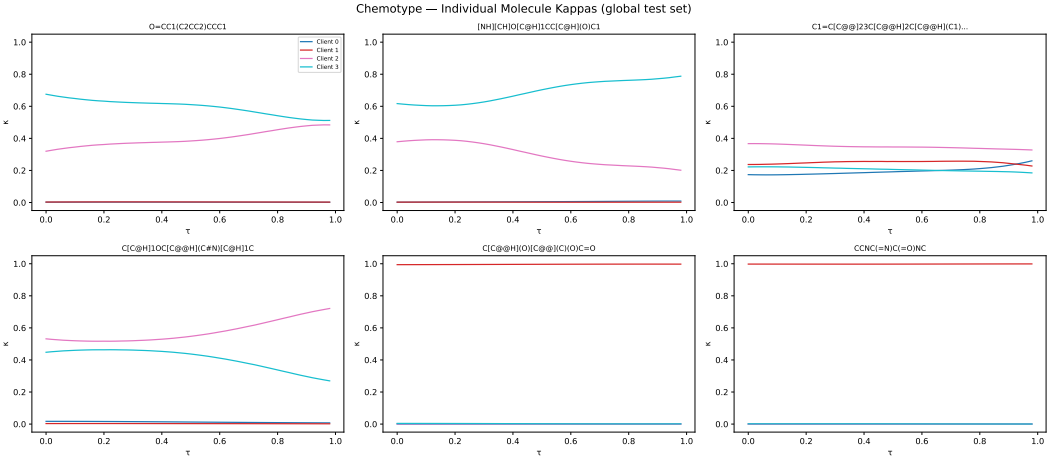


Figure 10: QM9 experiment: per-molecule $\kappa_k(\tau)$ trajectories from **Chemotype** split. Client trajectories mix on some molecules, for others one expert dominates.

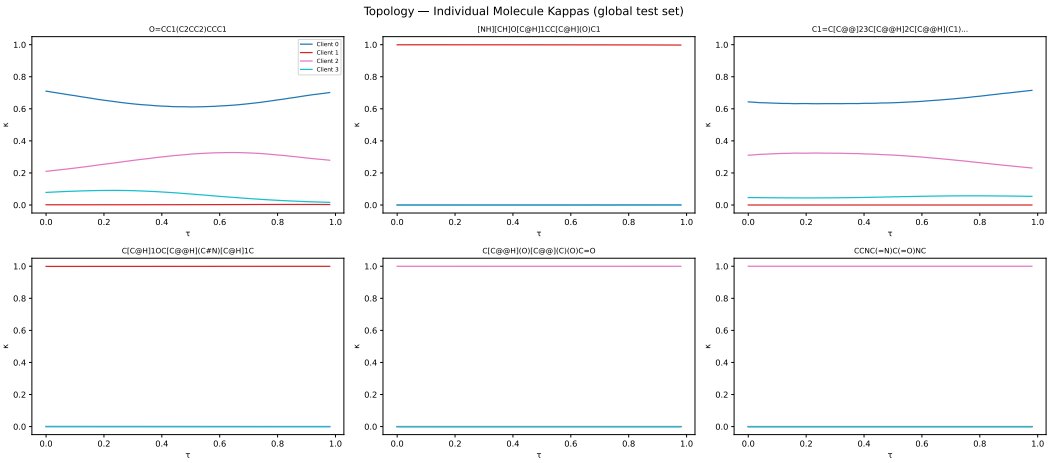


Figure 11: QM9 experiment: per-molecule $\kappa_k(\tau)$ trajectories from **Topology** split. A single expert dominates most of the shown trajectories.

Table 4: Full QM9 Topology local-step sweep. COV is in percent at $\delta = 0.50$ Å; MAT is in Å. There is one run per block size.

L	Merge rounds	COV-R ↑	MAT-R ↓	COV-P ↑	MAT-P ↓
1	50	94.972	0.1076	89.759	0.1381
2	25	94.996	0.1079	89.764	0.1383
5	10	94.999	0.1080	89.743	0.1388
10	5	94.966	0.1084	89.804	0.1393
25	2	95.175	0.1104	90.078	0.1443
50	1	95.165	0.1207	89.484	0.1669

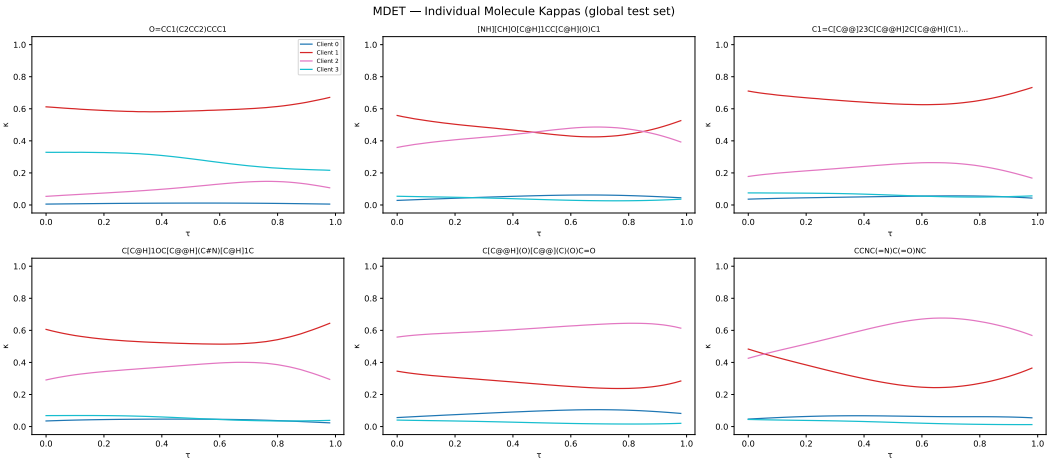


Figure 12: QM9 experiment: per-molecule $\kappa_k(\tau)$ trajectories from **MDET** split (labeled MLFF in the analysis exports).

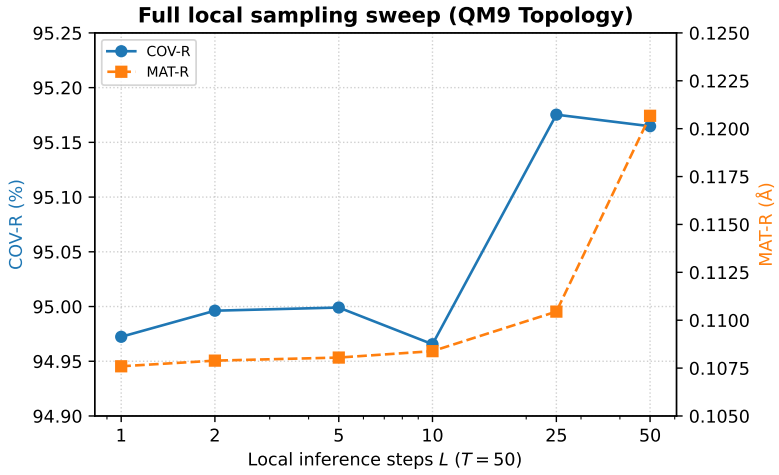


Figure 13: All tested local integration block sizes on QM9 Topology, with fixed $T = 50$. Each point represents one run on the same 996 molecules. Larger blocks reduce merging frequency but worsen matching distances at $L = 50$.