idea INTERNATIONAL DIGITAL ECONOMY ACADEMY
粤港澳大湾区数字经济研究院

The Thirty-Eighth Annual Conference
on Neural Information Processing Systems (**NeurIPS 2024**)

NEURAL INFORMATION PROCESSING SYSTEMS

SubgDiff: A Subgraph Diffusion Model to Improve Molecular Representation Learning

Jiying Zhang, Zijing Liu, Yu Wang, Bin Feng, Yu Li

International Digital Economy Academy (IDEA)

CODE

PAPER

Dec. 2024

The World Needs a Few Good IDEAs

Outline

- Background: diffusion model
- Motivation: molecular substructure
- SubgDiff: a subgraph diffusion model for molecular graph learning

Diffusion model (DDPM)

Forward Process (data→ noise)

$\mathbf{x}(0)$ $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon_{t-1}}$ $\mathbf{x}(T)$

(injects noise $\epsilon_{t-1} \in \mathcal{N}(0,1)$)

$\mathbf{x}(0)$ $\mathcal{L}_{simple} = \mathbb{E}_{t,x_0,\epsilon}[|\epsilon - \epsilon_\theta(x_t, \mathrm{t})|^2, \epsilon \sim \mathcal{N}(0,1)$ $\mathbf{x}(T)$

(denoising objective)

Reverse Process (noise → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." NeurIPS20

Figure from Song, Yang, et al. "Score-Based Generative Modeling through Stochastic Differential Equations." International Conference on Learning Representations.

Diffusion model

Forward Process (data→ noise)

$\mathbf{x}(0)$ $x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon_{t-1}}$ $\mathbf{x}(T)$

(injects noise $\epsilon_{t-1} \in \mathcal{N}(0,1)$)

$\mathbf{x}(0)$ $x_{i-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_i - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon_\theta}(x_t, t)\right) + \sigma_t z, \quad z \in \mathcal{N}(0,1)$ $\mathbf{x}(T)$

$\alpha_t = 1 - \beta_t, \ \bar{\alpha}_t = \Pi_{i=1}^{t}\alpha_i, \epsilon_\theta(x_t, t)$ is the noise predictor.

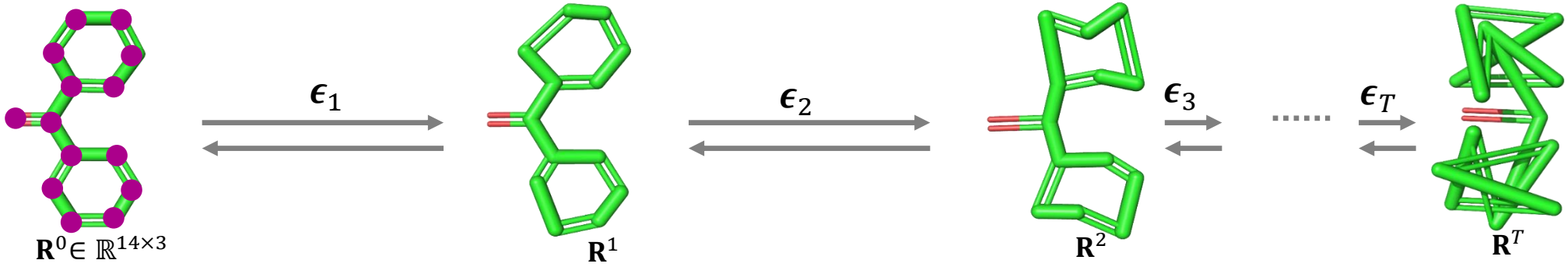
Sampling (noise x_T → data)

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." NeurIPS20

Diffusion model (DDPM) on 3D Molecules

Original 3D

Adding the noise to the position



$\mathbf{R}^i$: Atomic **coordinates** of 3D molecule

- **Forward process:**

$$\mathbf{R}^t = \sqrt{1-\beta_t}\mathbf{R}^{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t$$
$$= \sqrt{\bar{\alpha}_t}\mathbf{R}^0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$$

where $\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$.

Independently inject Gaussian noise into original 3D atomic position

- **Training (denoising) objective:** $\mathcal{L}_{simple} = \mathbb{E}_{t,\mathrm{R}^0,\epsilon}[|\epsilon - \epsilon_\theta(\mathbf{R}^t, \mathrm{t})|^2, \epsilon \sim \mathcal{N}(0,1)$

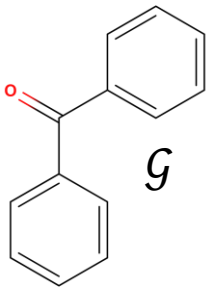
where $\epsilon_\theta(\mathbf{R}^t, \mathrm{t})$ is denoising networks, which can be used as molecule encoder.

- **Sampling:** $R^{t-1} = \frac{1}{\sqrt{\alpha_t}}(R^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(R^t, t)) + \sigma_t z, \quad z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

*Ho J, Jain A, Abbeel P. Denaoising Diffusion Probabilistic Models[J]. NeurIPS, 2020,

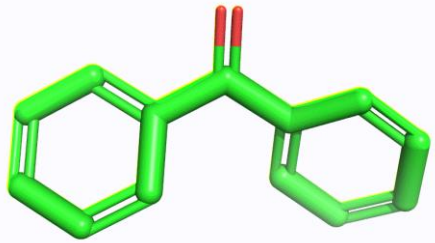
Molecular Representation Learning with Denoising

- **Conformation Generation**: get the 3D atomic Cartesian coordinates from the 2D molecular graph



2D molecular graph

Diffusion model[1]



3D molecular conformation

- Recent works[1,2] use this task as a pretraining technique for molecular representation learning.

★**Observation: Independently inject Gaussian noise** into original 3D atomic positions, **neglecting the substructure** in the molecules.

$$\mathbf{R}^t = \sqrt{1-\beta_t}\mathbf{R}^{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}_t$$

$$= \sqrt{\bar{\alpha}_t}\mathbf{R}^0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t$$

where $\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$.

Pre-training:

$$\mathcal{L}_{simple} = \mathbb{E}_{t,\mathrm{R}^0,\epsilon}\left[|\boldsymbol{\epsilon} - \boxed{\epsilon_\theta(\mathbf{R}^t, \mathcal{G}, \mathrm{t})}|^2, \boldsymbol{\epsilon} \sim \mathcal{N}(0,1)\right.$$

where $\epsilon_\theta(\mathbf{R}^t, \mathcal{G}, \mathrm{t})$ is the denoising network, which can be used as a molecular **encoder**.

Fine-tuning:

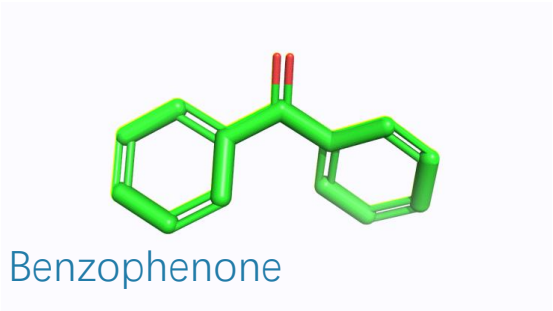
$$loss\left(f_{\theta'}\left(\boxed{\epsilon_\theta(\mathbf{R}^t, \mathcal{G}, \mathrm{t})}\right), y\right)$$

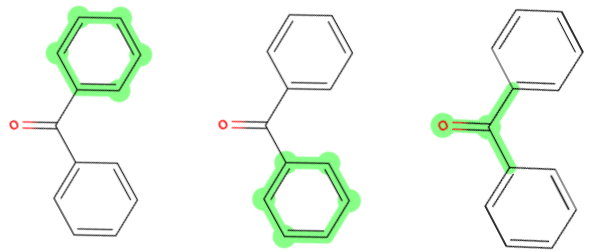
Where $f_{\theta'}$ denotes the prediction header.

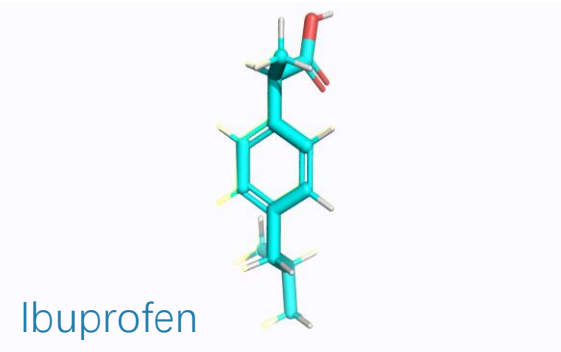
[1] Xu, Minkai, et al. "GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation." ICLR. 2021.
[2] Zaidi S, Schaarschmidt M, Martens J, et al. Pre-training via denoising for molecular property prediction[J]. ICLR, 2023.
[3] Liu, Shengchao, et al. "A group symmetric stochastic differential equation model for molecule multi-modal pretraining." ICML, 2023.

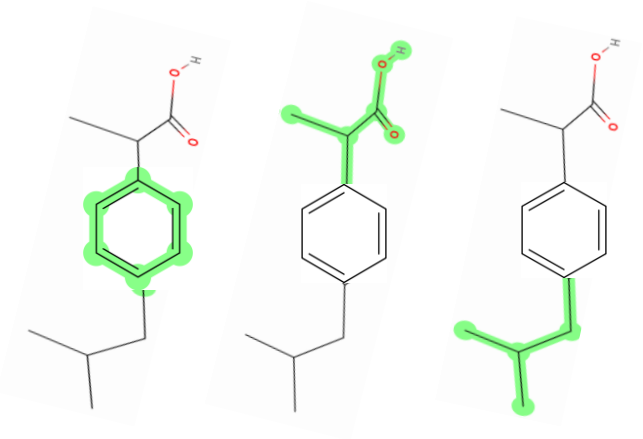
Molecular Substructure

- **Observation:** 3D **sub**structure substructures are closely related to molecular properties.
- **Decomposition approach**[1]: Torsional-based decomposition method

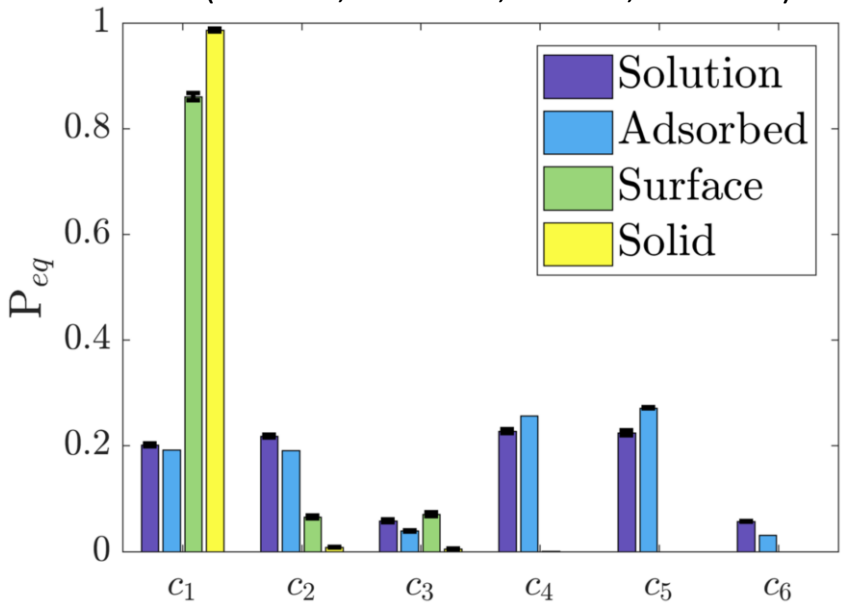


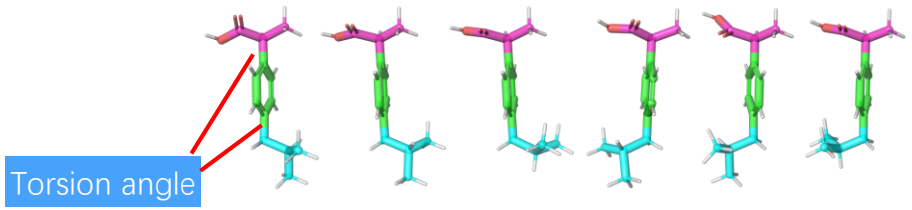


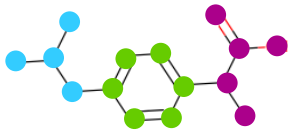




The equilibrium probability (p_{eq}) distribution of six different conformations (c1 to c6) of the **Ibuprofen** molecule in various environments (solution, adsorbed, surface, and solid)[2].







Three substructures

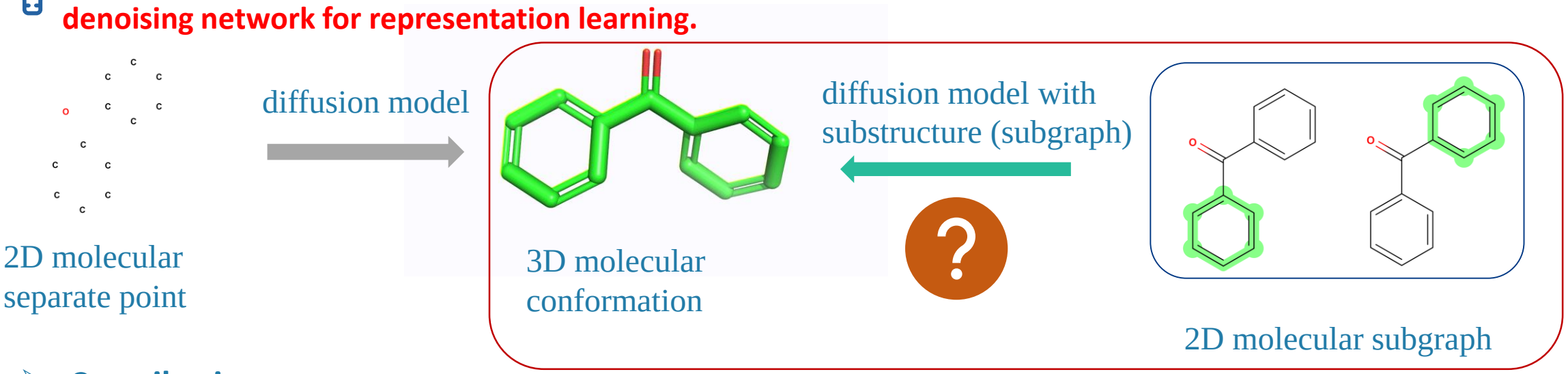
various torsion angles have different properties.

[1] Jing, Bowen, et al. "Torsional diffusion for molecular conformer generation." Advances in Neural Information Processing Systems 35 (2022): 24240-24253.
[2] Marinova, Veselina, et al. "Dynamics and thermodynamics of ibuprofen conformational isomerism at the crystal/solution interface." Journal of chemical theory and computation (2018)

Motivation

- Existing diffusion models on molecules **independently inject Gaussian noise** into atomic coordinates during the forward process, **neglecting the substructure** in the molecules which plays a significant role in molecular representation learning

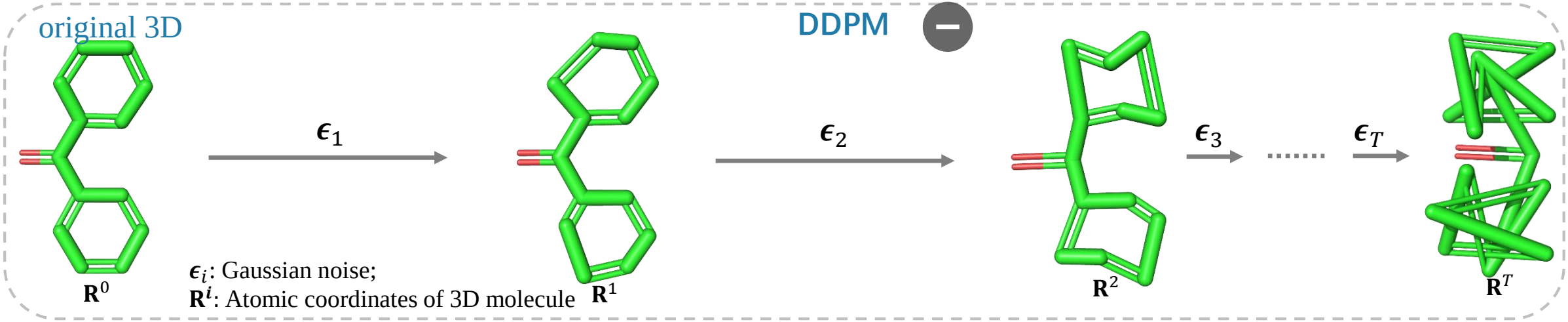
It remains open to exploring the molecular **substructure in Diffusion model, to improve the denoising network for representation learning.**



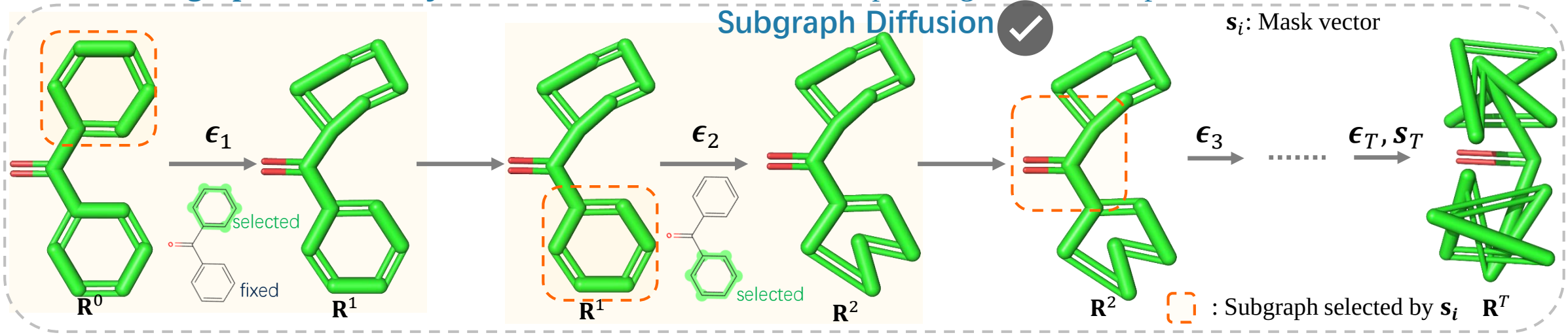
- **Contribution：**
  - Incorporate the **substructure information** into diffusion models to improve molecular representation learning;
  - Propose a new diffusion model **SubgDiff** that adopts subgraph prediction, expectation state and k-step same-subgraph diffusion to improve its sampling and training;

Forward process: SubgDIFF vs DDPM

Independently inject Gaussian noise into original 3D atomic position



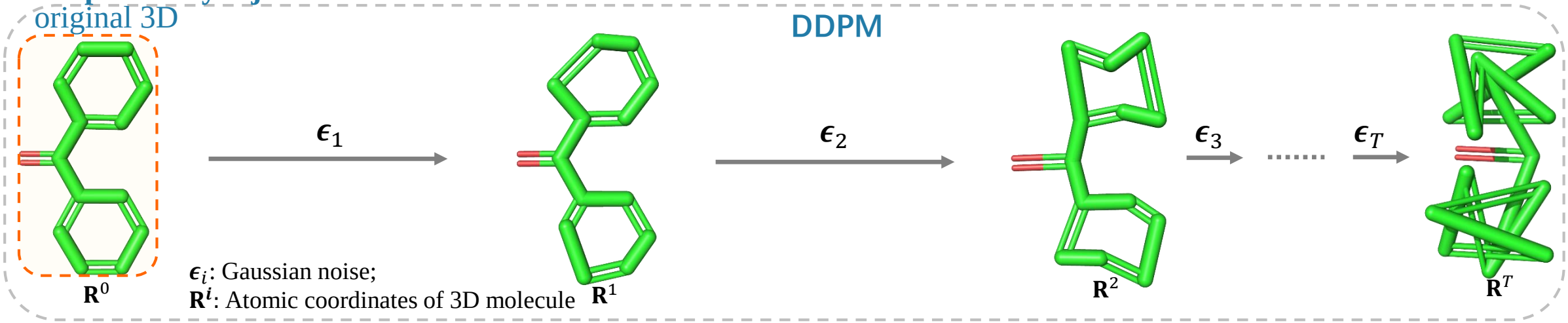
Select a subgraph and then inject Gaussian noise into the corresponding substructure position

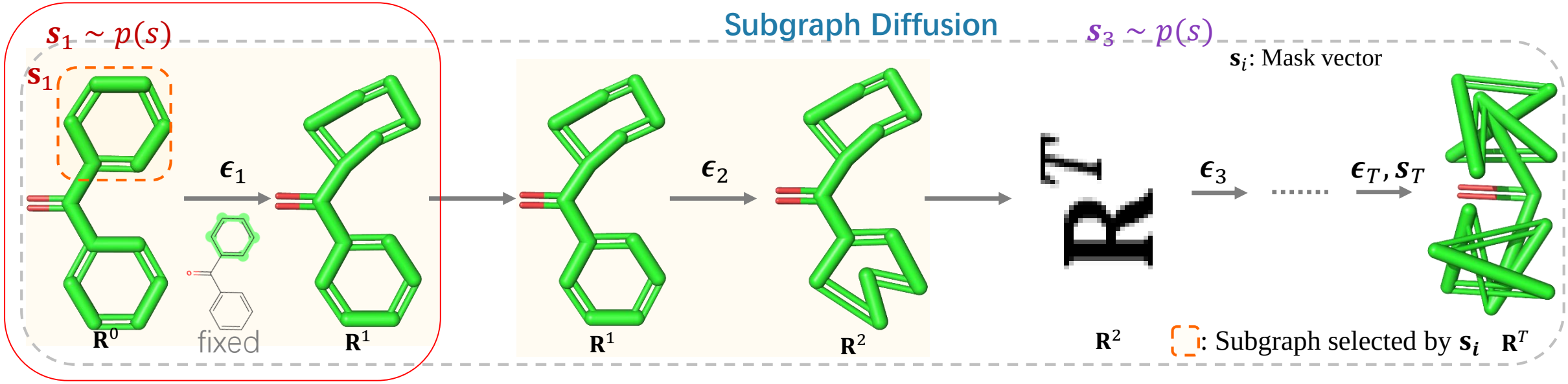


*Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[J]. NeurIPS, 2020,

Forward process: SubgDiff vs DDPM

independently inject Gaussian noise into atoms

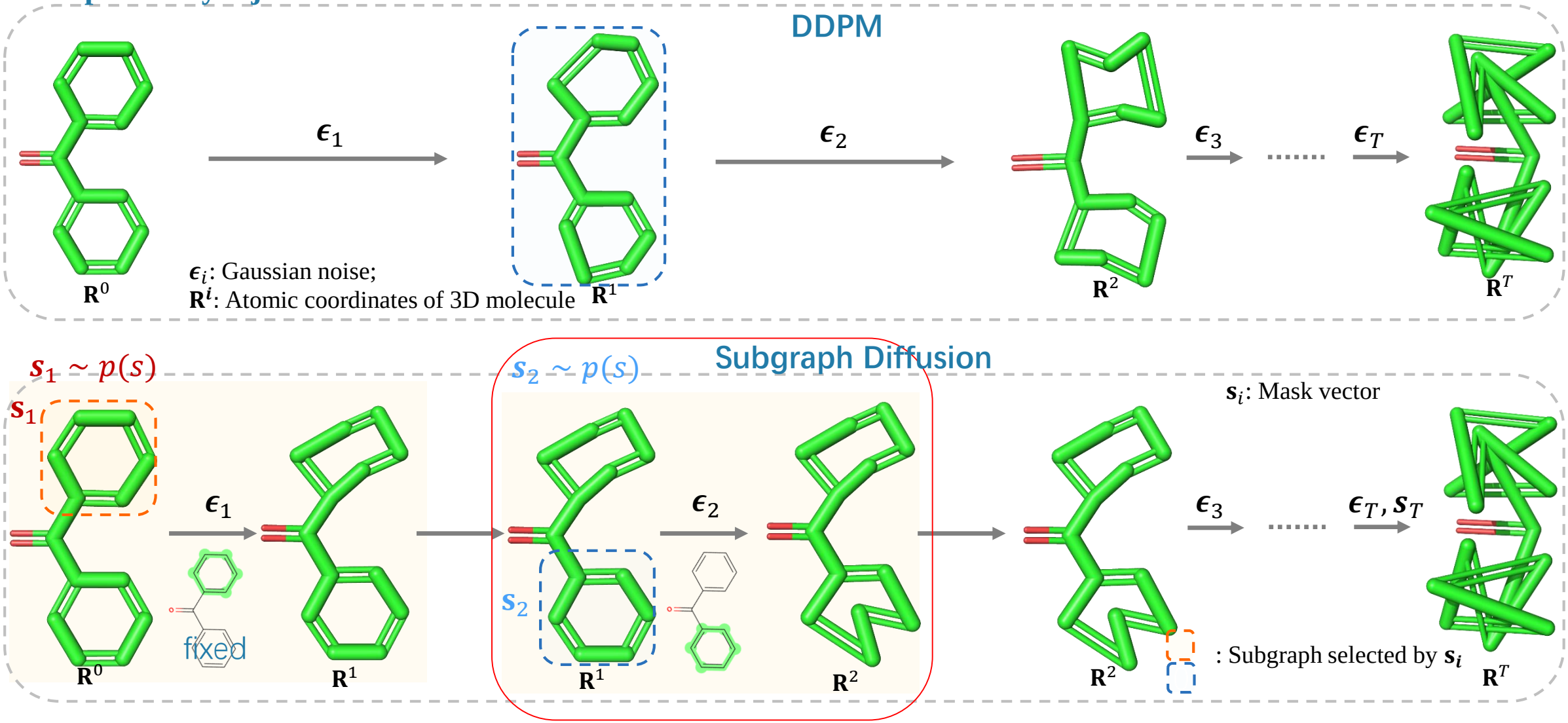




*Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[J]. NeurIPS, 2020,

Forward process: SubgDiff vs DDPM

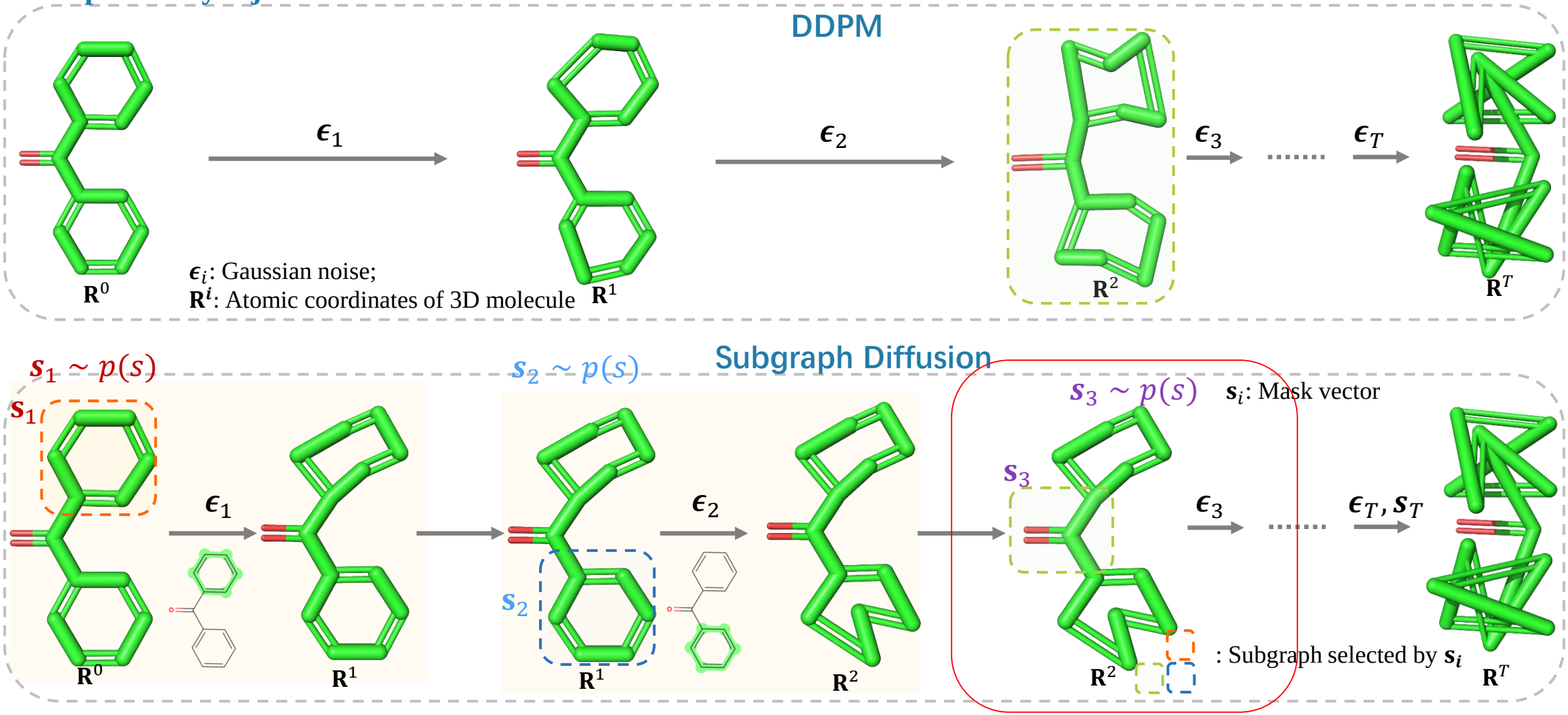
independently inject Gaussian noise into atoms



*Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[J]. NeurIPS, 2020,

Forward process: SubgDiff vs DDPM

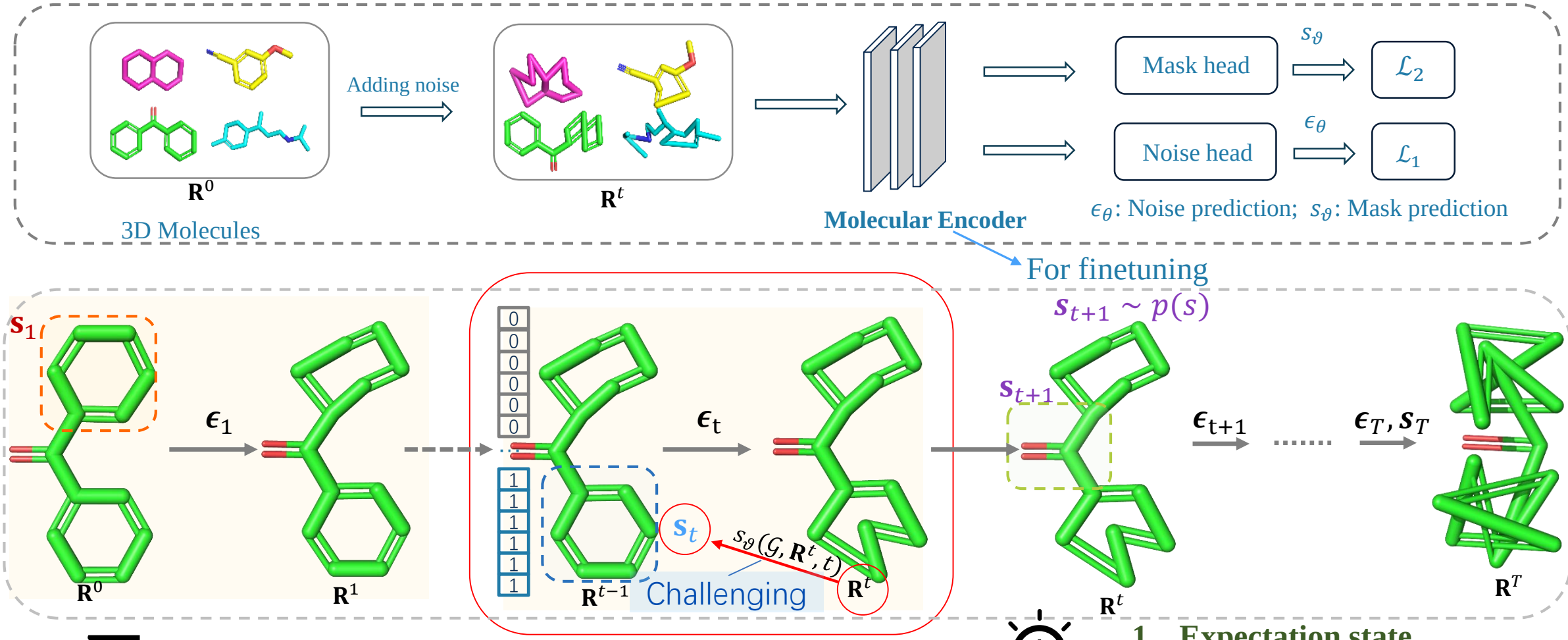
independently inject Gaussian noise into atoms



*Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models[J]. NeurIPS, 2020,

Training objective

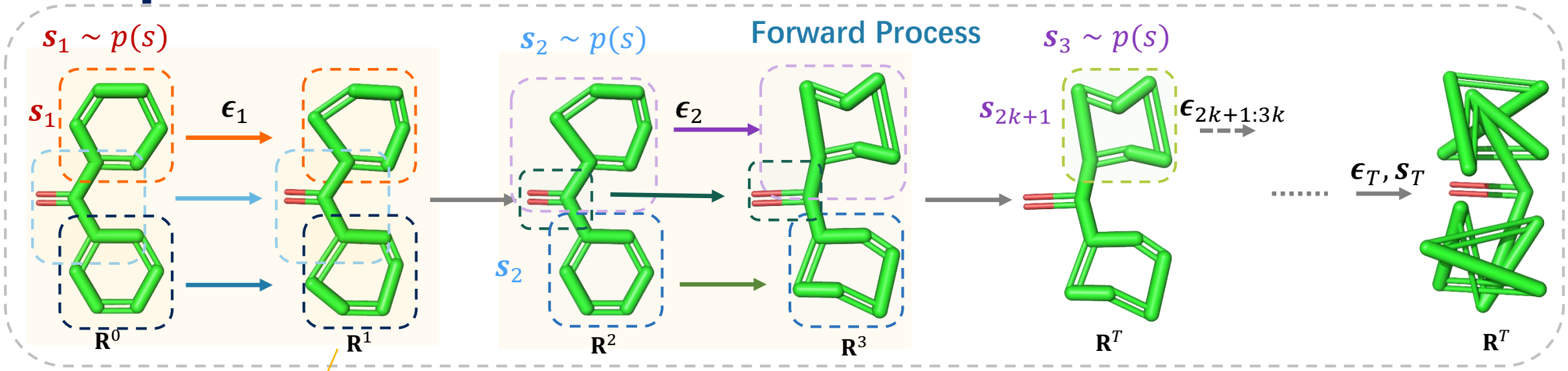
$$\mathcal{L}_{simple}(\theta,\vartheta) = \mathbb{E}_{t,\mathbf{R}^0,\mathbf{s}_t,\epsilon}[\underbrace{\| \operatorname{diag}(\mathbf{s}_t)(\epsilon - \epsilon_\theta(\mathcal{G},\mathbf{R}^t,t) \|^2}_{\mathcal{L}_1 \text{ noise prediction loss}} + \underbrace{\lambda \mathrm{BCE}(\mathbf{s}_t, s_\vartheta(\mathcal{G},\mathbf{R}^t,t))}_{\mathcal{L}_2 \text{ subgraph prediction loss}}]$$

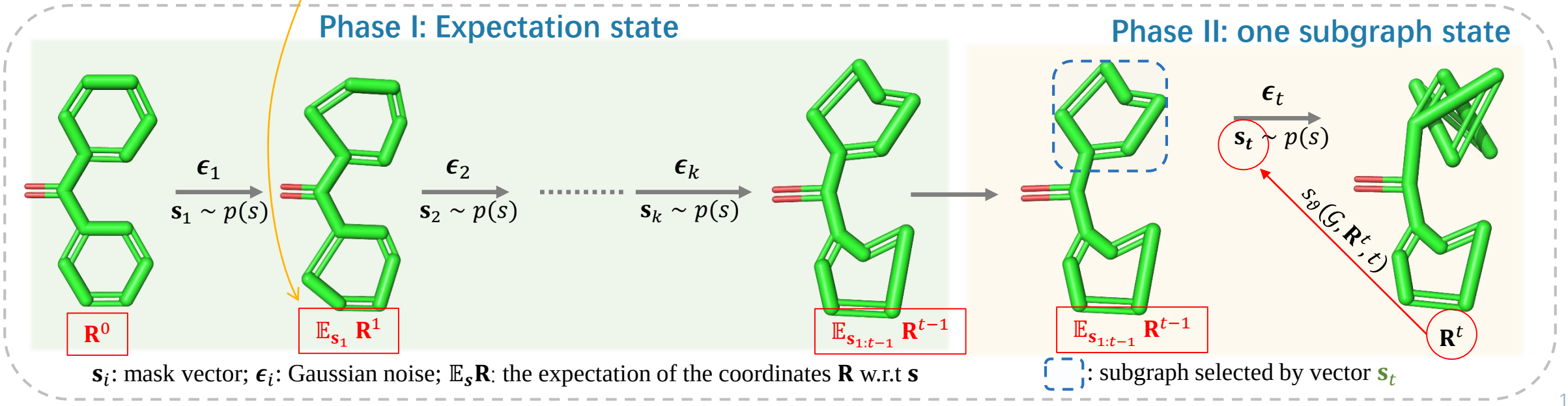


Challenge: subgraph prediction loss $\mathcal{L}_2$ is difficult to converge.

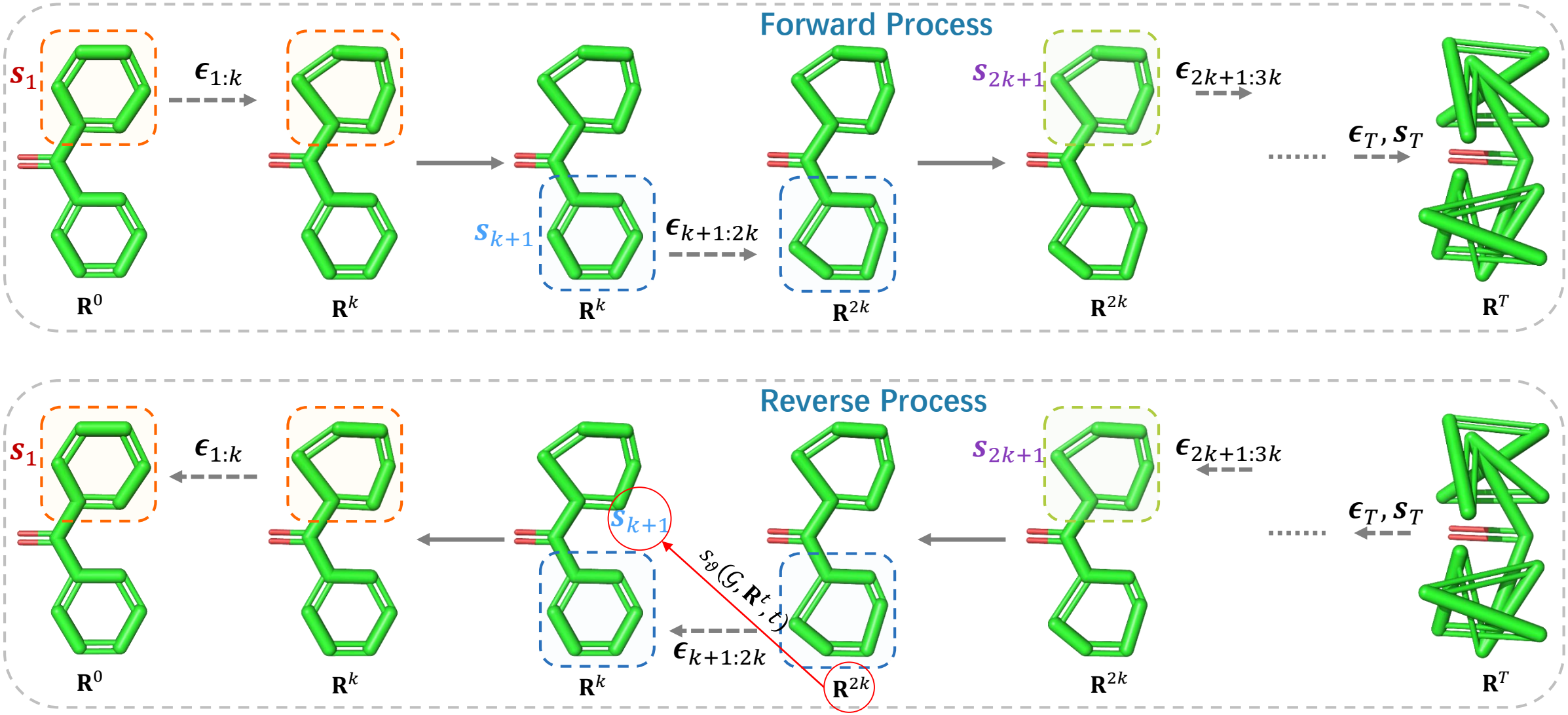
1. **Expectation state**
2. **k-step same subgraph diffusion**

Expectation State

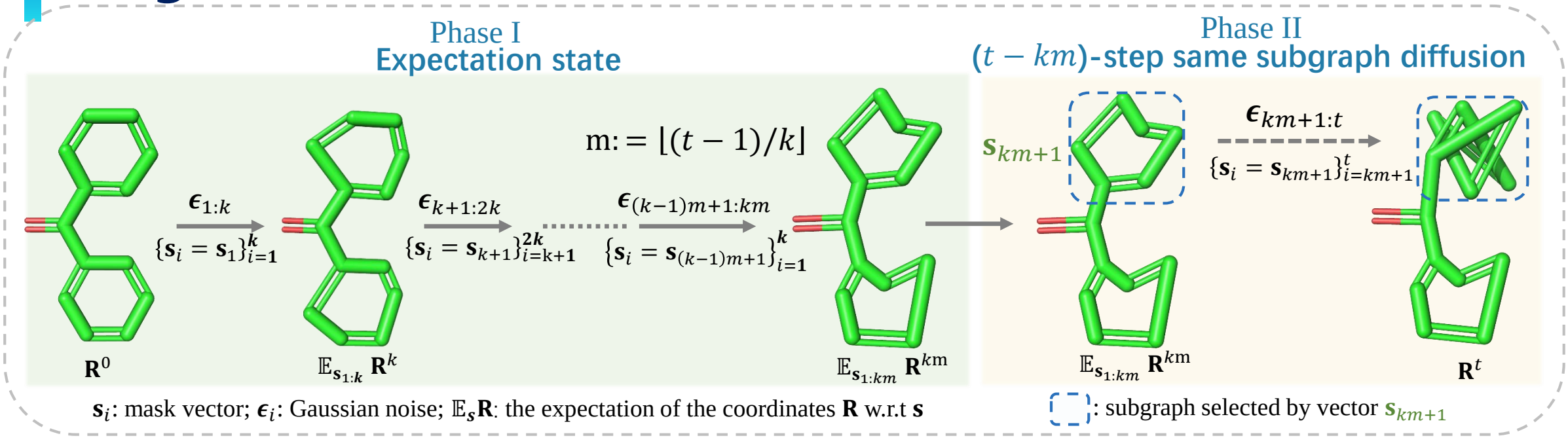




SubgDiff with k-step same subgraph diffusion



SubgDiff state transition



Phase I: $\mathbb{E}_{s_{1:km}} R^{km} = \sqrt{\bar{\alpha}_m}\, R^0 + p\sqrt{\sum_{l=1}^{m} \frac{\bar{\alpha}_m}{\bar{\alpha}_l}\left(1 - \prod_{i=(l-1)k+1}^{kl}(1-\beta_i)\right)}\, \epsilon_0$, where $\alpha_j = \left(p\sqrt{\prod_{i=(j-1)k+1}^{k=j}(1-\beta_i)} + 1 - p\right)^2$.

$\mathrm{m} := \lfloor t/k \rfloor$ applies the floor function to the quotient t/k, rounding it down to the nearest integer.

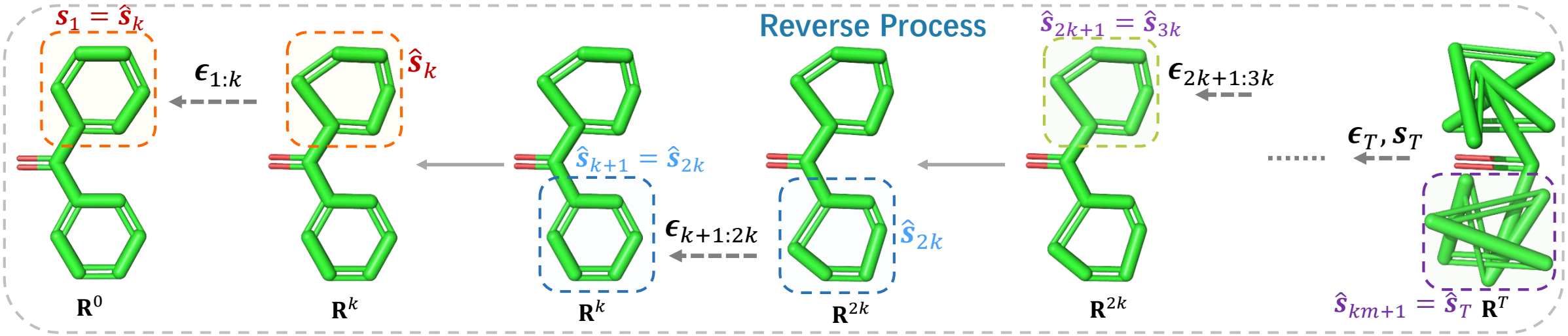
Phase II: $R^t = \sqrt{\prod_{i=km+1}^{t}(1-\beta_i s_{km+1})}\, \mathbb{E}_{s_{1:km}} R^{km} + \sqrt{1 - \prod_{i=km+1}^{t}(1-\beta_i s_{km+1})}\, \epsilon_{km}$.

$R^0 \to R^t$: $q(R^t|R^0, s_{km+1}) = \mathcal{N}\left(R^t; \sqrt{\frac{\bar{\gamma}_t \bar{\alpha}_m}{\bar{\gamma}_{km}}} R^0, \left(\frac{\bar{\gamma}_t}{\bar{\gamma}_{km}} p^2 \sum_{l=1}^{m} \frac{\bar{\alpha}_m}{\bar{\alpha}_l}\left(1 - \frac{\bar{\beta}_{kl}}{\bar{\beta}_{(l-1)k}}\right) + 1 - \frac{\bar{\gamma}_t}{\bar{\gamma}_{km}}\right) I\right)$,

where p is the selection probability of each node, $\gamma_i = 1 - \beta_i s_{km+1}$, $\bar{\gamma}_t = \prod_{i=1}^{t} \gamma_i$, and $\bar{\beta}_t = \prod_{i=1}^{t} (1-\beta_i)\}$

Sampling

$$R^{t-1} = \frac{1}{\sqrt{\gamma_t}}\left(R^t - \frac{\hat{s}_{km+1}\beta_t}{\sqrt{\gamma_t\delta + \hat{s}_{km+1}\beta_t}}\epsilon_\theta(R^t, t)\right) + \frac{\sqrt{\hat{s}_{km+1}\beta_t}\sqrt{\frac{\bar{\gamma}_{t-1}}{\bar{\gamma}_{km}}p^2\sum_{l=1}^{m}\frac{\bar{\alpha}_m}{\bar{\alpha}_l}\left(1-\frac{\bar{\beta}_{kl}}{\bar{\beta}_{(l-1)k}}\right)+1-\frac{\bar{\gamma}_{t-1}}{\bar{\gamma}_{km}}}}{\sqrt{\gamma_t\delta + \hat{s}_{km+1}\beta_t}}z, \quad (19)$$



SubgDiff Framework

Algorithm 1: Training SubgDiff

Input: A molecular graph G_{3D}, k for same mask diffusion, $m := \lfloor (t-1)/k \rfloor$

Sample $t \sim \mathcal{U}(1, ..., T)$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

Sample $\mathbf{s}_{km+1} \sim p_{s_{km+1}}(\mathcal{S} \mid \mathcal{G})$ ▷ Sample a subgraph

$\mathbf{R}^t \leftarrow q(\mathbf{R}^t | \mathbf{R}^0)$ ▷ Equation 17

$\mathcal{L}_1 = \mathrm{BCE}(\mathbf{s}_{km+1}, s_\vartheta(\mathcal{G}, \mathbf{R}^t, t))$ ▷ Subgraph prediction loss

$\mathcal{L}_2 = \|\mathrm{diag}(\mathbf{s}_{km+1})(\epsilon - \epsilon_\theta(\mathcal{G}, \mathbf{R}^t, t))\|^2$ ▷ Denoising loss

$\mathrm{optimizer.step}(\mathbb{E}_{t, \mathbf{R}^0, \mathbf{s}_t, \epsilon}[\lambda \mathcal{L}_1 + \mathcal{L}_2])$ ▷ Optimize parameters θ, ϑ

Algorithm 2: Sampling from SubgDiff

k is the same as training, for k-step same-subgraph diffusion;

Sample $\mathbf{R}^T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ Random noise initialization

for $t = T$ **to** 1 **do**

  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$ ▷ Random noise

  If $t\%k == 0$ or $t == T$: $\hat{\mathbf{s}} \leftarrow s_\vartheta(\mathcal{G}, \mathbf{R}^t, t)$ ▷ Subgraph prediction

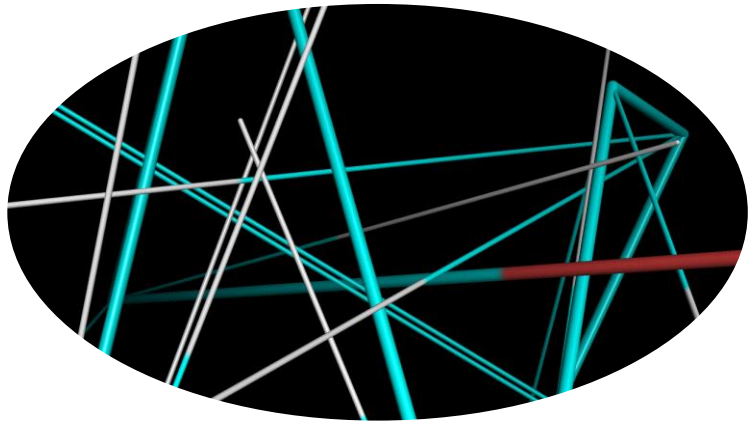
  $\hat{\epsilon} \leftarrow \epsilon_\theta(\mathcal{G}, \mathbf{R}^t, t)$ ▷ Posterior

  $\mathbf{R}^{t-1} \leftarrow$ Equation 19 ▷ sampling

end

return $\mathbf{R}^0$

SubgDiff Conformation generation



$$\text{COV-R}(S_g, S_r) = \frac{1}{|S_r|}\left|\left\{\mathcal{C} \in S_r \,|\, \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}) \leq \delta, \hat{\mathcal{C}} \in S_g\right\}\right|,$$

$$\text{MAT-R}(S_g, S_r) = \frac{1}{|S_r|}\sum_{\mathcal{C} \in S_r} \min_{\hat{\mathcal{C}} \in S_g} \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}),$$

$$\text{COV-P}(S_r, S_g) = \frac{1}{|S_g|}\left|\left\{\hat{\mathcal{C}} \in S_g \,|\, \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}) \leq \delta, \mathcal{C} \in S_r\right\}\right|,$$

$$\text{MAT-P}(S_r, S_g) = \frac{1}{|S_g|}\sum_{\hat{\mathcal{C}} \in S_g} \min_{\mathcal{C} \in S_r} \text{RMSD}(\mathcal{C}, \hat{\mathcal{C}}),$$

Table 4: Results on **GEOM-QM9** dataset under different diffusion timesteps. DDPM (Ho et al., 2020) is the sampling method used in GeoDiff. Our proposed sampling method (Algorithm 2) can be viewed as a DDPM variant. ▲/▼ denotes SubgDiff outperforms/underperforms GeoDiff. The threshold $\delta = 0.5$Å.

			COV-R (%) ↑		MAT-R (Å) ↓		COV-P (%) ↑		MAT-P (Å) ↓	
Models	Timesteps	Sampling method	Mean	Median	Mean	Median	Mean	Median	Mean	Median
GeoDiff	5000	DDPM	80.36	83.82	0.2820	0.2799	53.66	50.85	0.6673	0.4214
SubgDiff	5000	DDPM (ours)	90.91▲	95.59▲	0.2460▲	0.2351▲	50.16▼	48.01▼	0.6114▲	0.4791▼
GeoDiff	500	DDPM	80.20	83.59	0.3617	0.3412	45.49	45.45	1.1518	0.5087
SubgDiff	500	DDPM (ours)	89.78▲	94.17▲	0.2417▲	0.2449▲	50.03▲	48.31▲	0.5571▲	0.4921▲
GeoDiff	200	DDPM	69.90	72.04	0.4222	0.4272	36.71	33.51	0.8532	0.5554
SubgDiff	200	DDPM (ours)	85.53▲	88.99▲	0.2994▲	0.3033▲	47.76▲	45.89▲	0.6971▲	0.5118▲

SubgDiff finetuning on MD17 (3D)

Pretrain on PCQM4Mv2. The backbone is SchNet.

Table 12: Results on eight **force** prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is mean absolute error, and the best results are marked in bold and underlined, respectively.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
– (random init)	1.203	0.380	0.386	0.794	0.587	0.826	0.568	0.773
Type Prediction	1.383	0.402	0.450	0.879	0.622	1.028	0.662	0.840
Distance Prediction	1.427	0.396	0.434	0.818	0.793	0.952	0.509	1.567
Angle Prediction	1.542	0.447	0.669	1.022	0.680	1.032	0.623	0.768
3D InfoGraph	1.610	0.415	0.560	0.900	0.788	1.278	0.768	1.110
RR	1.215	0.393	0.514	1.092	0.596	0.847	0.570	0.711
InfoNCE	1.132	0.395	0.466	0.888	0.542	0.831	0.554	0.664
EBM-NCE	1.251	0.373	0.457	0.829	0.512	0.990	0.560	0.742
3D InfoMax	1.142	0.388	0.469	0.731	0.785	0.798	0.516	0.640
GraphMVP	1.126	0.377	0.430	0.726	0.498	0.740	0.508	0.620
GeoSSL-1L	1.364	0.391	0.432	0.830	0.599	0.817	0.628	0.607
GeoSSL	1.107	0.360	0.357	0.737	0.568	0.902	0.484	0.502
MoleculeSDE (VE)	1.112	0.304	0.282	0.520	0.455	0.725	0.515	0.447
MoleculeSDE (VP)	1.244	0.315	0.338	0.488	0.432	0.712	0.478	0.468
Ours	**0.880**	**0.252**	**0.258**	**0.459**	**0.325**	**0.572**	**0.362**	**0.420**

SubgDIFF finetuning on MoleculeNet (2D)

Pretrain on PCQM4Mv2. The backbone is GIN.

Table 2: Results for 2D molecular property prediction tasks (with 2D topology only). We report the mean (and standard deviation) ROC-AUC of three random seeds with scaffold splitting for each downstream task. The backbone is GIN. The best and second best results are marked bold and underlined, respectively.

Pre-training	BBBP ↑	Tox21 ↑	ToxCast ↑	Sider ↑	ClinTox ↑	MUV ↑	HIV ↑	Bace ↑	Avg ↑
– (random init)	68.1±0.59	75.3±0.22	62.1±0.19	57.0±1.33	83.7±2.93	74.6±2.35	75.2±0.70	76.7±2.51	71.60
AttrMask	65.0±2.36	74.8±0.25	62.9±0.11	61.2±0.12	<u>87.7±1.19</u>	73.4±2.02	76.8±0.53	79.7±0.33	72.68
ContextPred	65.7±0.62	74.2±0.06	62.5±0.31	<u>62.2±0.59</u>	77.2±0.88	75.3±1.57	77.1±0.86	76.0±2.08	71.28
InfoGraph	67.5±0.11	73.2±0.43	63.7±0.50	59.9±0.30	76.5±1.07	74.1±0.74	75.1±0.99	77.8±0.88	70.96
MolCLR	66.6±1.89	73.0±0.16	62.9±0.38	57.5±1.77	86.1±0.95	72.5±2.38	76.2±1.51	71.5±3.17	70.79
3D InfoMax	68.3±1.12	76.1±0.18	64.8±0.25	60.6±0.78	79.9±3.49	74.4±2.45	75.9±0.59	79.7±1.54	72.47
GraphMVP	69.4±0.21	76.2±0.38	64.5±0.20	60.5±0.25	86.5±1.70	76.2±2.28	76.2±0.81	79.8±0.74	73.66
MoleculeSDE(VE)	68.3±0.25	76.9±0.23	<u>64.7±0.06</u>	60.2±0.29	80.8±2.53	<u>76.8±1.71</u>	77.0±1.68	79.9±1.76	73.15
MoleculeSDE(VP)	<u>70.1±1.35</u>	<u>77.0±0.12</u>	64.0±0.07	60.8±1.04	82.6±3.64	76.6±3.25	<u>77.3±1.31</u>	<u>81.4±0.66</u>	<u>73.73</u>
Ours	**70.2±2.23**	**77.2±0.39**	**65.0±0.48**	**62.2±0.97**	**88.2±1.57**	**77.3±1.17**	**77.6±0.51**	**82.1±0.96**	**74.85**

THANK YOU

www.idea.edu.cn

IDEA Official WeChat