GOLD: Geometric Optimized Latent Diffusion for Structure-Aware RNA Inverse Folding

Qi Si[1*] **Xuyang Liu**[1*] **Penglei Wang**[2*] **Shuo Su**[3] **Limei Han**[4]
Mengzhu Li[4] **Xin Guo**[1,4†] **Yuan Qi**[1,4,5] **Yuan Cheng**[1,4†]

[1]Shanghai Academy of Artificial Intelligence for Science
[2]School of Biomedical Engineering, Shanghai Jiao Tong University
[3]Community Science Department, Kuaishou
[4]Artificial Intelligence Innovation and Incubation Institute, Fudan University
[5]Zhongshan Hospital, Fudan University

Abstract

Designing RNA sequences that fold into specific 3D structures is a fundamental challenge in therapeutics. Current RNA inverse folding methods typically condition on 3D backbone coordinates but the generation is limited to 1D discrete sequences. This ignores the continuous side-chain orientations, which are essential for overall structural stability. To address this issue, we explicitly introduce side-chain geometry into the RNA generative process. Unlike co-design frameworks that diffuse across separate raw spaces, we propose Geometric Optimized Latent Diffusion (**GOLD**), which project sequence and structure into a unified latent manifold. GOLD employs a novel GeoVAE to elegantly integrate discrete RNA sequences with continuous side-chain angles. By mapping these modalities into a single, smooth latent manifold, GOLD naturally facilitates a synergistic co-diffusion process. To enhance structural validity, we further introduce a training-free, gradient-guided sampling mechanism. By decoding the generated sequence probabilities and side-chain geometries into structures during the sampling process, we compute analytical gradients from physical constraints. This dynamic feedback actively guides the generated sequence to a physically stable state. Extensive experiments validate the superiority of GOLD, achieving state-of-the-art performance across sequence recovery, secondary structure validity, and 3D foldability metrics. Our code and data are publicly available at `https://github.com/darkflash03/GOLD`.

1 Introduction

RNA is essential in modern therapeutics and synthetic biology, driving key innovations such as mRNA vaccines and RNA aptamers [12, 54]. Since the biological function of RNA molecules depends largely on their 3D structures, the development of new RNAs requires the design of sequences that can reliably fold into specific target structures. This task of designing a nucleotide sequence for a target 3D structure is known as RNA inverse folding [13, 18]. Driven by deep learning, current generative models typically condition on the 3D coordinates of the RNA backbone to predict discrete tokens in 1D sequence space [48, 28]. However, the spatial orientations of nucleobase side-chains directly dictate the formation of base-pairing and stacking interactions in RNA, which ultimately govern the stability of the entire 3D structure [31, 36]. Due to this geometric dependency, there are inherent limitations in restricting the generation of targets to 1D sequence space. If the orientation

*contributed equally to this work.
†Corresponding to `guoxin@sais.org.cn` and `cheng_yuan@fudan.edu.cn`.

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

of the side chain in 3D space is not explicitly modeled, the generated sequences may fail to satisfy spatial constraints, resulting in atomic collisions within the generated structure.

Beyond 1D sequence generation, a broader paradigm in macromolecular modeling attempts to jointly capture sequences and 3D geometries. Pioneered in protein engineering [33, 10] and recently extended to **de novo** RNA design [35, 32], these frameworks highlight the importance of modeling both modalities. However, a fundamental distinction exists in the task objective: while **de novo** co-design generates both sequence and structure from scratch, inverse folding operates under the physical constraints of a predefined 3D backbone. To optimally leverage side-chain geometry within this rigidly constrained setting, we propose Geometric Optimized Latent Diffusion (GOLD). GOLD employs a Geometry-aware Variational Autoencoder (GeoVAE) to elegantly map discrete RNA sequences and continuous side-chain angles into a unified, smooth latent manifold. We further refine this manifold by aligning it with embeddings from a pre-trained RNA foundation model, leveraging its sequence representations to enrich our latent space. By executing the diffusion process directly within this shared and enriched space, our model inherently couples sequence identity with 3D geometry, ensuring a natural and consistent alignment of physical constraints during sequence generation.

In addition to a unified representation, maintaining the physical consistency of the generated structures is essential. To achieve this goal, current state-of-the-art models [45, 23] often utilize Reinforcement Learning (RL) [5, 16, 53] to fine-tune their output based on external folding tools, which typically requires significant computational resources. To provide a more efficient alternative, GOLD leverages explicit sequence and continuous side-chain angles to introduce a training-free, gradient-guided sampling mechanism. During the denoising process, our method calculates gradients directly from 2D pairing rules and 3D collision constraints to guide the sequence toward physically stable states without the need for additional post-training.

Our main contributions are summarized as follows:

- **Explicit Geometric Awareness in RNA Inverse Folding:** We identify continuous side-chain orientation as the critical missing link in current inverse folding methods, and introduce continuous side-chain orientations as a generative target in our method.
- **Unified Representation for RNA Co-generation:** We introduce a novel generative framework that combines GeoVAE and Latent Diffusion Model (LDM). GeoVAE compresses discrete sequences and continuous side-chain angles into a unified latent manifold, enabling the LDM to perform physically consistent sequence generation.
- **Training-Free Gradient Guidance:** Based on our continuous geometric representation, we introduce a sampling mechanism that incorporates analytical gradients from 2D pairing and 3D collision constraints.
- **Comprehensive Benchmark Performance:** GOLD achieves state-of-the-art results across a comprehensive suite of 1D, 2D, and 3D metrics. Our model simultaneously improves sequence recovery and physical foldability, demonstrating that explicit geometric modeling leads to more reliable RNA designs.

2 Related Work

2.1 RNA Inverse Folding

Historically, computational modeling of RNA inverse folding began as a thermodynamic optimization problem. Early physics-based methods [21, 59, 8, 2] framed sequence design as a task to minimize free-energy through dynamic programming and heuristic search. As the field advanced, deep learning architectures began to replace traditional algorithms. Initial data-driven strategies primarily relied on reinforcement learning and autoregressive models [41]. To better capture structural dependencies, the field subsequently shifted to Graph Neural Networks (GNNs) [48, 28, 49], which actively leverage spatial message passing in 3D coordinates. Concurrently, RNA foundation models [9, 38, 7] were developed to capture deep co-evolutionary semantics. Most recently, following breakthroughs in protein design [26, 56], diffusion frameworks were adapted for RNA design, including discrete generative diffusion [24] and hyperbolic discrete diffusion [22]. To enhance structural fidelity, reinforcement learning [45, 23] is currently integrated to fine-tune these diffusion models based on external folding rewards. However, these refinement approaches typically require significant computational requirements.

2.2 Latent Diffusion and Controllable Generation

Latent Diffusion Models (LDMs) manage the complexity of discrete or mixed data spaces by compressing information into continuous, low-dimensional manifolds, based on foundational autoencoders [30, 19, 52]. Operating within a continuous diffusion framework [20, 47] provides a smooth and differentiable optimization landscape while reducing computational overhead. This formulation enables controllable generation through gradient-based guidance [11, 57, 60], which guide the reverse sampling trajectory by injecting gradients from auxiliary energy functions in each denoising step. Based on these advances, GOLD maps discrete RNA sequences and side-chains into a shared continuous latent space. This representation supports a training-free sampling process, using gradients from 2D pairing and 3D collision constraints to guide the model toward physically valid structures.

3 Preliminaries

3.1 Geometric Representation of RNA

Without explicit awareness of physical volume, RNA inverse folding models frequently produce discrete sequences that suffer from steric clashes when translated into 3D structures (Figure 1). Structurally, the RNA nucleobases exhibit a rigid, planar geometry inherent to their aromatic rings [1]. Since sequence identity typically defines this internal shape, the nucleobase's motion can be effectively approximated as a single rotation around the glycosidic bond. Consequently, this limitation establishes the glycosidic torsion angle χ as the primary structural degree of freedom (DoF) defining the spatial orientation and collision boundaries of the side-chains [3].

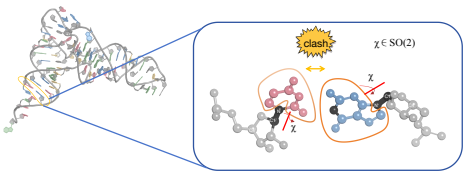

Figure 1: **RNA geometry.** The backbone establishes local frames ($\mathbf{R}_{bb}$), while the side-chain orientation is governed by the glycosidic torsion angle ($\chi \in \mathrm{SO}(2)$). Modeling this angle in 3D space helps identify potential steric clashes between rigid base planes.

To formalize this constraint, we note that while the global position and orientation of a side-chain are represented in $\mathrm{SE}(3)$, its DoF under the rigid-body assumption is restricted to a 1D rotation around the glycosidic bond axis, represented by $\mathrm{SO}(2)$. Here, we define the Kinematic Approximation Error (**KAE**) as the root-mean-square deviation (RMSD) between the native atomic coordinates of the side-chain and those recovered jointly using the standard rigid base templates, the backbone coordinates and the side-chain angles χ. To evaluate this physical prior, we analyzed approximately 8,000 native RNA structures. Our results demonstrate that retaining only the angle $\chi \in \mathrm{SO}(2)$ as a 1D continuous DoF is sufficient to bound the KAE below 0.2 Å.

Based on this insight, we define a deterministic $\mathrm{SE}(3)$-equivariant kinematic building operator $\mathcal{B}$. Given the coarse-grained coordinates of the backbone $\mathbf{X}_{bb} \in \mathbb{R}^{N\times 3\times 3}$, we extract local rotation matrices $\mathbf{R}_{bb}^{(i)} \in \mathrm{SO}(3)$ and translation vectors $\mathbf{T}_{bb}^{(i)} \in \mathbb{R}^3$ from anchor atoms. Assuming standard rigid-body templates $\mathbf{T}_{base}^{(i)} \in \mathbb{R}^{4\times 14\times 3}$, the side-chain coordinates $\mathbf{X}_{sc}^{(i)}$ can be reconstructed as:

$$\mathbf{X}_{sc}^{(i)} = \mathcal{B}(\mathbf{S}_i, \chi_i, \mathbf{X}_{bb}^{(i)}) = (\mathbf{S}_i \cdot \mathbf{T}_{base})\, \mathbf{R}_{\chi_i} \mathbf{R}_{bb}^{(i)} + \mathbf{T}_{bb}^{(i)} \tag{1}$$

where $\mathbf{S}_i$ is the predicted sequence probability and $\mathbf{R}_{\chi_i} \in \mathrm{SO}(3)$ denotes the 3D rotation matrix constructed from the 1D angle $\chi_i \in \mathrm{SO}(2)$ around the local glycosidic bond axis. By formalizing this geometric relationship, we not only capture the intrinsic coupling between sequence probabilities and side-chain geometry but also make $\mathcal{B}$ as a differentiable operator. This lays the groundwork for active physical interventions: for instance, when spatial overlaps occur during generation, $\mathcal{B}$ translates physical repulsion into analytical gradients that backpropagate to refine both sequence probabilities and torsional orientations. We detail this mechanism in Appendix A.

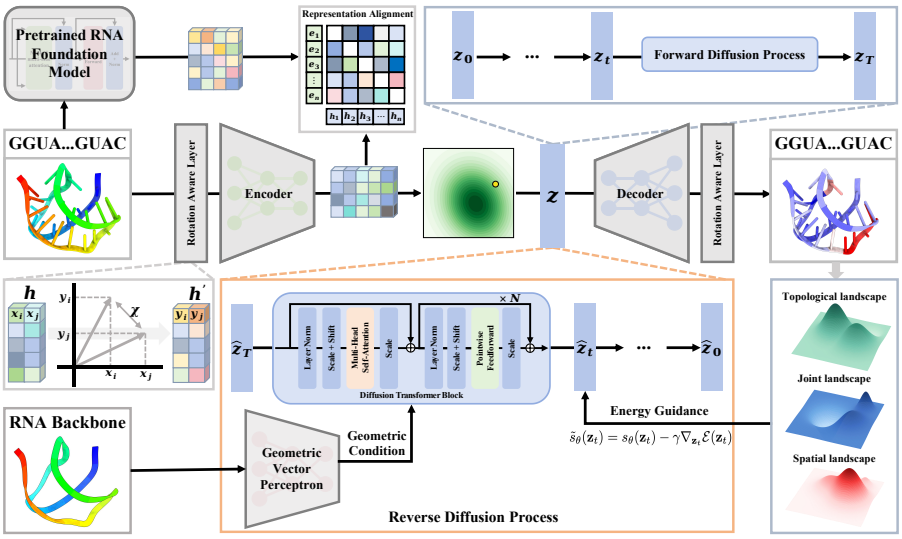

Figure 2: GOLD architecture for RNA inverse folding task. GOLD first encodes RNA sequences and structures into a unified latent space using an rotation aware (SO(2)) encoder, then decodes the latent embeddings into the raw space. Subsequently, a score-based diffusion model, enhanced by a geometric vector perceptron module, is employed in the latent space for RNA generation. During inference, GOLD utilizes an energy-based guidance mechanism to generate desired RNA samples.

3.2 Problem Formulation

Given a target RNA backbone $\mathbf{X}_{bb}$ of length N, our primary objective is to generate a physically valid discrete sequence $\mathbf{S} \in \mathcal{A}^N$, where $\mathcal{A} = \{\text{A, U, C, G}\}$ is the nucleotide alphabet. Traditional inverse folding models often treat this as a purely token prediction task [25]. This limits their ability to perceive geometric information such as side-chain collisions. As the analysis above suggests, the physical validity of a sequence is significantly influenced by the side chain torsion angle χ. Therefore, the core problem we seek to address is: **How can side-chain angles serve as geometric constraints to guide physically valid sequence generation?**

4 Methods

4.1 Overall Framework

As illustrated in Fig 2, we propose GOLD, a geometry-aware latent diffusion framework for RNA inverse folding. GOLD consists of three stages: (1) a Geometric Variation Autoencoder (**GeoVAE**) maps discrete sequences and torsion angles into a continuous latent space; (2) a latent diffusion model captures the distribution of valid RNA designs conditioned on 3D backbone geometry; and (3) a differentiable energy-guided sampling procedure guides reverse denoising to generate spatially feasible and topologically consistent samples.

4.2 Geometric Variational Autoencoder (GeoVAE)

To process heterogeneous sequence tokens and geometric angles while enabling interaction with the 3D physical space, we design an $\mathrm{SO}(2)$-equivariant GeoVAE.

Rotation-Aware Encoding and Decoding. Let the dense features of the discrete sequence $\mathbf{S} \in \{0,1\}^{N\times 4}$ be $\mathbf{H} \in \mathbb{R}^{N\times D}$. To eliminate 2π periodicity, continuous torsion angles $\boldsymbol{\chi}$ are encoded as unit circle vectors $c_i = [\cos\chi_i, \sin\chi_i]^\top$. As shown in the bottom left of Figure 2, during encoding process, the **Rotation Aware Layer** reshapes sequence features $\mathbf{h}_i$ into complex pairs and applies

$\mathrm{SO}(2)$ rotations:

$$\begin{bmatrix} x'_{i,k} \\ y'_{i,k} \end{bmatrix} = \mathbf{R}(\chi_i) \begin{bmatrix} x_{i,k} \\ y_{i,k} \end{bmatrix} = \begin{bmatrix} \cos\chi_i & -\sin\chi_i \\ \sin\chi_i & \cos\chi_i \end{bmatrix} \begin{bmatrix} x_{i,k} \\ y_{i,k} \end{bmatrix} \tag{2}$$

Following this rotation-aware integration, the coupled features pass through multiple Transformer encoder layers, which ultimately map the inputs into the continuous latent distribution $q_\phi(\mathbf{z}|\mathbf{S}, \boldsymbol{\chi}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. During decoding, the sampled latent variable $\mathbf{z}$ is decoupled through the rotation-aware layer [55, 50]:

- **Sequence**: We decode the probability using the magnitude: $\hat{\mathrm{s}}_i = \mathrm{MLP}_{seq}(\sqrt{x^2 + y^2})$.
- **Torsion**: To maintain sensitivity to spatial orientation, the angle is decoded directly from the complete phase information: $\hat{\chi}_i = \mathrm{Normalize}(\mathrm{MLP}_{angle}(\mathbf{z}_i))$.

Joint Optimization. We feed the predicted soft probabilities $\hat{\mathbf{S}}$ and angles $\hat{\boldsymbol{\chi}}$ directly into the operator $\mathcal{B}$ defined in Equation (1). Combined with the fixed target backbone $\mathbf{X}_{bb}$, this will output the 3D coordinates of the side chain $\hat{\mathbf{X}_{\mathbf{sc}}}$. This operation establishes a differentiable pathway from the latent space $\mathcal{Z}$ to the physical space. The GeoVAE is optimized end-to-end via the following joint objective:

$$\mathcal{L}_{GeoVAE} = \lambda_1 \mathcal{L}_{seq} + \lambda_2 \mathcal{L}_{angle} + \lambda_3 \mathcal{L}_{coord} + \lambda_4 \mathcal{L}_{repa} + \lambda_5 \mathcal{L}_{KL} \tag{3}$$

where each term provides explicit supervision: **(1)** $\mathcal{L}_{seq}$ **(Cross-Entropy)** minimizes the discrepancy between the predicted soft probabilities $\hat{\mathbf{S}}$ and the ground-truth one-hot sequence $\mathbf{S}$; **(2)** $\mathcal{L}_{angle}$ **(MSE)** computes the reconstruction error between the predicted continuous 2D torsion vectors and the native vectors; **(3)** $\mathcal{L}_{coord}$ **(MSE)** directly penalizes the spatial deviation between the reconstructed side-chain coordinates $\hat{\mathbf{X}}_{sc}$ and the native side-chain coordinates $\mathbf{X}_{sc}$; **(4)** $\mathcal{L}_{repa}$ **(Cosine Similarity)** computes $1-\cos(\mathbf{z}, \mathbf{z}_{FM})$ to align the latent variable $\mathbf{z}$ with the embeddings of pre-trained foundation model $\mathbf{z}_{FM}$ [58]; and **(5)** $\mathcal{L}_{KL}$ **(KL Divergence)** regularizes the approximate posterior $q_\phi(\mathbf{z}|\mathbf{S}, \boldsymbol{\chi})$ to conform to the standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

4.3 Latent Diffusion

After establishing the continuous manifold $\mathcal{Z}$, the sequence generation task is formulated as a diffusion process within this continuous space, avoiding the complex transition matrix optimization typically required in discrete diffusion.

Forward and Reverse Diffusion Processes. Given a clean latent representation $\mathbf{z}_0 \sim \mathcal{Z}$, the forward diffusion process progressively injects Gaussian noise according to a predefined noise schedule $\bar{\alpha}_t$:

$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1 - \bar{\alpha}_t)\mathbf{I}) \tag{4}$$

During the reverse denoising process, the generative distribution is conditioned on the target backbone $\mathbf{X}_{bb}$. We construct a spatial graph and utilize Geometric Vector Perceptrons (GVP) [27] to extract rotation-invariant scalars S and $\mathrm{SE}(3)$-equivariant vectors $\mathbf{V}$:

$$\mathbf{S}' = \mathrm{MLP}_s\Big(\big[s \parallel \|\mathbf{W}_h\mathbf{V}\|_2\big]\Big), \quad \mathbf{V}' = \mathbf{W}_v(\mathbf{W}_h\mathbf{V}) \odot \sigma\big(\mathrm{MLP}_{gate}(s')\big) \tag{5}$$

where $\mathbf{W}_h$ and $\mathbf{W}_v$ are learnable parameters. As illustrated in Figure 2, these updated representations (S, V) collectively constitute the **Geometric Condition**, providing the essential 3D structural scaffold to modulate the subsequent diffusion process.

Model Optimization. These aggregated geometric conditions, along with timestep embeddings, modulate the hidden states of the Diffusion Transformer (DiT) [37] via AdaLN mechanisms. With the GeoVAE parameters frozen, the DiT is trained to predict the clean latent $\mathbf{z}_0$ by minimizing the mean squared error:

$$\mathcal{L}_{Diff} = \mathbb{E}_{\mathbf{z}_0,\epsilon,t}\left[\|\mathbf{z}_\theta(\mathbf{z}_t, \mathbf{X}_{bb}, t) - \mathbf{z}_0\|_2^2\right] \tag{6}$$

4.4 Dual-Guided Sampling

Although the diffusion model effectively captures the global data distribution, generating physically feasible RNAs requires explicitly resolving conflicts to ensure collision-free and topologically correct structures. Leveraging the differentiable pathway established by operator $\mathcal{B}$, we introduce a dual-guided mechanism to actively direct the sampling process.

Energy Guidance from Bayesian Perspective. In conditional diffusion models, the conditional score function sampled from the posterior distribution $p(\mathbf{z}_t|c)$ can be decoupled by Bayes' theorem into a prior and a guidance term [15]:

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t|c) = \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t) + \nabla_{\mathbf{z}_t} \log p(c|\mathbf{z}_t) \tag{7}$$

To explicitly inject physical interventions, we model the constraint c as a Boltzmann distribution governed by an energy function $\mathcal{E}$, such that $p(c|\mathbf{z}_t) \propto \exp(-\mathcal{E}(\mathbf{z}_t))$. Evaluating the logarithmic gradient of this term and substituting it into the Bayesian formulation yields the modified score function (as illustrated at the bottom of the Figure 2):

$$\tilde{s}_\theta(\mathbf{z}_t) = s_\theta(\mathbf{z}_t) - \gamma \nabla_{\mathbf{z}_t} \mathcal{E}(\mathbf{z}_t) \tag{8}$$

where γ is the guidance scale. This mechanism demonstrates that subtracting the energy gradient pushes the denoising trajectory towards local minima that conform to defined constraints.

Construction of Energy Landscapes. To address the issue of spatial crowding, we define the **Spatial landscape** $\mathcal{E}_{clash}$ on the reconstructed coordinates $\hat{\mathbf{X}}_{\mathbf{sc}}$, imposing a van der Waals repulsive penalty on non-adjacent atom pairs with distances below the threshold τ:

$$\mathcal{E}_{clash} = \frac{1}{2} \sum_{a,b \notin \text{adjacent}} \max(0, \tau - D_{a,b})^2 \tag{9}$$

Simultaneously, given paired indices $(i, j) \in \mathcal{E}_{pair}$ from the target 2D graph, the **Topological landscape** $\mathcal{E}_{ss}$ is defined as the negative log-likelihood based on the valid pair set $\mathcal{V}$ to ensure adherence to base-pairing rules:

$$\mathcal{E}_{ss} = -\frac{1}{|\mathcal{E}_{pair}|} \sum_{(i,j)\in\mathcal{E}_{pair}} \log\Big(\sum_{(A,B)\in\mathcal{V}} \hat{\mathbf{P}}_{i,A} \hat{\mathbf{P}}_{j,B} + \epsilon \Big) \tag{10}$$

where $\hat{\mathbf{P}}_{i,A}$ is the decoded probability of nucleotide A at position i. At each denoising step t, we compute the **Joint landscape** $\mathcal{E}_{total} = \omega_{ss}\mathcal{E}_{ss} + \omega_{clash}\mathcal{E}_{clash}$ and backpropagate the gradients. This mechanism guides the model to simultaneously fine-tune both the soft sequence probabilities and the χ angles, thereby achieving zero-shot generalization to physical constraints. For a detailed explanation of GeoVAE, latent diffusion, and the dual-guided sampling process, including complete algorithmic pseudocode and specific model hyperparameter configurations, please refer to Appendix B.

5 Experiments

5.1 Experimental Setup

Datasets. We construct the **GOLD Benchmark**, a high-quality, non-redundant RNA inverse folding dataset sourced from the PDB. After rigorous dual-stage clustering to prevent data leakage, the final dataset contains $7,583$ training, $1,144$ validation, and 598 testing samples. For out-of-distribution (OOD) evaluation on highly complex targets, we incorporate the RNA subset from the **CASP15** challenge [14]. Detailed construction protocols are provided in Appendix C.1.

Baselines. We benchmark GOLD against state-of-the-art autoregressive and diffusion models for inverse folding: PiFold [17], RDesign [48], gRNAde [28], RiboDiffusion [24], and SOLD [45]. Baseline implementation specifics are detailed in Appendix C.2.

Evaluation Metrics. To ensure fair comparisons against 1D-only baselines, we employ external oracles to evaluate 1D **Sequence Recovery**, 2D Secondary Structure (**SS**) Recovery (via RNAfold [21]), and 3D structural **RMSD** (via AlphaFold3 [1] and RhoFold [44]). Furthermore, to validate the physical of GOLD's **native** geometry without relying on black-box refolding tools, we define two intrinsic metrics for ablation studies: **SS Validity** (topological pairing success rate) and **Clash Score** (steric collisions per 1,000 atoms). All formal mathematical definitions are detailed in Appendix C.3.

5.2 Main Results

We evaluate GOLD against several representative methods on both our internal GOLD Benchmark and the challenging out-of-distribution CASP15 targets. As shown in Table 1, our full model (**GOLD**

Table 1: Performance comparison of 3D RNA inverse folding methods. Evaluation is conducted on the internal GOLD Benchmark and the external CASP15 targets. Results are reported as mean ± standard deviation over five random seeds.

Method	Seq Recovery (↑)	SS Recovery (↑)	RMSD (Å, ↓)	
			RhoFold	AlphaFold3
GOLD Benchmark (In-Distribution)				
PiFold	0.4123 ± 0.0354	0.6184 ± 0.0648	17.26 ± 3.56	18.73 ± 4.27
RDesign	0.4430 ± 0.0326	0.6751 ± 0.1106	14.65 ± 4.36	15.12 ± 3.85
gRNAde	0.5453 ± 0.0873	0.6450 ± 0.1183	15.38 ± 4.84	16.10 ± 4.37
RiboDiffusion	0.5148 ± 0.0574	0.6948 ± 0.0846	13.07 ± 3.19	13.89 ± 3.92
SOLD	$\underline{0.5843 \pm 0.0635}$	$\underline{0.7366 \pm 0.0983}$	$\underline{12.61 \pm 2.79}$	$\underline{13.65 \pm 4.06}$
GOLD (Base)	0.6249 ± 0.0283	0.7785 ± 0.0782	11.99 ± 2.84	13.26 ± 4.34
GOLD (Dual-Guided)	$\mathbf{0.6380 \pm 0.0292}$	$\mathbf{0.8186 \pm 0.0724}$	$\mathbf{10.65 \pm 2.63}$	$\mathbf{12.31 \pm 3.63}$
CASP15 (Out-of-Distribution)				
PiFold	0.3156 ± 0.0362	0.3407 ± 0.0564	36.23 ± 4.94	35.43 ± 6.72
RDesign	0.3640 ± 0.0305	0.4250 ± 0.0568	33.18 ± 4.86	38.42 ± 5.86
gRNAde	0.5161 ± 0.0204	0.4266 ± 0.0478	33.89 ± 5.53	39.23 ± 5.39
RiboDiffusion	0.4959 ± 0.0265	0.6134 ± 0.0505	31.14 ± 3.57	30.74 ± 4.28
SOLD	$\underline{0.5877 \pm 0.0106}$	$\underline{0.6367 \pm 0.0481}$	$\underline{28.87 \pm 3.23}$	$\underline{27.74 \pm 4.04}$
GOLD (Base)	0.6187 ± 0.0246	0.6521 ± 0.0484	28.32 ± 3.76	27.17 ± 4.74
GOLD (Dual-Guided)	$\mathbf{0.6624 \pm 0.0308}$	$\mathbf{0.7089 \pm 0.0427}$	$\mathbf{27.64 \pm 3.57}$	$\mathbf{26.18 \pm 3.93}$

(Dual-Guided)), consistently achieves the best performance in all evaluation metrics. Specifically, it demonstrates robust improvements in 1D sequence recovery and 2D topological accuracy, while yielding more physically feasible 3D structures with lower RMSD scores (as evaluated by both RhoFold and AlphaFold3). Notably, even in the absence of explicit guidance, **GOLD (Base)** remains highly competitive and consistently outperforms the strongest baseline (SOLD). This result further underscores the inherent effectiveness of our unified sequence-structure latent space in capturing critical structural dependencies.

Impact of the Unified Sequence-Structure Manifold. As observed in Table 1, the better performance of SOLD compared to RiboDiffusion highlights the benefits of projecting discrete sequences into a continuous latent space. Moreover, our unguided baseline **GOLD (Base)**, achieves further improvements, outperforming SOLD across all metrics on the GOLD Benchmark (e.g., sequence recovery of 0.6249 vs. 0.5843). We attribute this performance gain primarily to the difference in representation spaces. SOLD relies on a unimodal sequence latent space, whereas GOLD further introduces the **joint modeling of sequences and side-chain angles**. By encoding discrete sequence tokens and continuous torsion angles into an $SO(2)$-equivariant manifold, GOLD provides an optimization landscape for diffusion models that is inherently capable of capturing 3D physical dependencies.

Impact of Dual-Guided Sampling. Although the unified latent space provides a strong structural prior, relying solely on this latent representation may not fully resolve fine-grained physical constraints. To mitigate this issue, we introduce a dual-guided mechanism into the sampling process. In our experiments, we set $\omega_{ss} = 1.0$ to enforce topological pairing and $\omega_{clash} = 0.1$ to balance spatial constraints. As shown in Table 1, on the GOLD benchmark, this dual-guided mechanism improves the 2D topological pairing accuracy (SS Recovery) from 0.7785 to 0.8186 over the unguided baseline, while further reducing the AlphaFold3-based RMSD from 13.26 Å to 12.31 Å. Unlike methods that require further fine-tuning, this dual-guided mechanism is completely training-free.

Generalization on CASP15 Targets. The CASP15 dataset provides a out-of-distribution (OOD) test involving highly complex, unseen RNA conformations. As Table 1 shows, even without explicit guidance, GOLD (Base) outperforms the strongest baseline (SOLD) in all metrics. Performance can be further enhanced by employing dual-guided sampling (GOLD), resulting in optimal overall performance, including a sequence recovery rate of 0.6624 and an AlphaFold3 RMSD of 26.18Å. Ultimately, these results demonstrate that GOLD performs strongly in zero-shot generalization: the

joint sequence and side-chain angle modeling provides a strong foundational advantage on novel topologies and dual-guided sampling further enhances this advantage.

5.3 Ablation Studies

To systematically analyze the contributions of key components in GOLD, we conduct further ablation experiments on the internal GOLD Benchmark dataset.

Explicit Sequence–Torsion Modeling. Unlike conventional RNA inverse-folding methods that primarily use geometric features as structural conditions, GOLD explicitly models the glycosidic torsion angle χ alongside nucleotide identity in a unified latent representation. To evaluate the contribution of explicit side-chain torsion modeling, we introduce **GOLD (Seq-only)**, a controlled variant that removes the $SO(2)$-aware fusion module and the geometric losses $\mathcal{L}_{\text{angle}}$ and $\mathcal{L}_{\text{coord}}$. Unlike SOLD, which also performs sequence-only latent diffusion but employs a different VAE, GOLD (Seq-only) retains the same latent dimension, RNA-FM alignment, backbone-conditioned GVP, diffusion backbone, and training settings as GOLD (Base). Both variants are evaluated without gradient guidance to isolate the effect of joint sequence–torsion modeling. Importantly, GOLD (Seq-only) still incorporates backbone geometry as structural conditioning, allowing us to distinguish explicit torsion modeling from the conventional use of geometric features. As shown in Table 2, GOLD (Seq-only) performs comparably to SOLD across sequence recovery, secondary-structure recovery, and RhoFold-based 3D evaluation, whereas GOLD (Base) consistently outperforms both sequence-only variants. These results demonstrate that the improvements of GOLD are not solely attributable to latent diffusion or foundation-model alignment, but benefit from explicitly coupling sequence identity with side-chain torsion geometry in a unified latent representation.

Table 2: Ablation of explicit sequence–torsion modeling on the GOLD Benchmark. All methods are evaluated without gradient guidance.

Method	**Torsion χ**	**Seq Recovery (↑)**	**SS Recovery (↑)**	**RhoFold RMSD (Å, ↓)**
SOLD	×	0.5843 ± 0.0635	0.7366 ± 0.0983	12.61 ± 2.79
GOLD (Seq-only)	×	0.5897 ± 0.0346	0.7402 ± 0.1036	12.74 ± 2.93
GOLD (Base)	✓	$\mathbf{0.6249 \pm 0.0283}$	$\mathbf{0.7785 \pm 0.0782}$	$\mathbf{11.99 \pm 2.84}$

Impact of Geometric Supervision. To further investigate the contribution of geometric supervision in GeoVAE, we separately remove the angle loss $\mathcal{L}_{\text{angle}}$ and coordinate loss $\mathcal{L}_{\text{coord}}$, while keeping the remaining architecture and training settings unchanged. As shown in Table 3, removing $\mathcal{L}_{\text{angle}}$ mainly degrades torsion prediction accuracy, while $\mathcal{L}_{\text{coord}}$ still provides indirect geometric supervision. In contrast, removing $\mathcal{L}_{\text{coord}}$ substantially increases coordinate reconstruction error and KAE, despite maintaining relatively accurate torsion predictions. These results demonstrate that the two geometric losses play complementary roles, and accurate torsion prediction alone is insufficient to ensure geometrically consistent 3D reconstruction.

Table 3: Ablation of geometric supervision losses in GeoVAE on the GOLD Benchmark.

Variant	**Angle MSE (↓)**	**Coord MSE (↓)**	**KAE (Å, ↓)**
w/o $\mathcal{L}_{\text{angle}}$	0.0546	0.0388	0.2802
w/o $\mathcal{L}_{\text{coord}}$	0.0136	0.1611	0.6812
Full GeoVAE	**0.0109**	**0.0287**	**0.2466**

Impact of Alignment on Foundation Model Representation. In Table 4, we evaluate the impact of Foundation Model (FM) alignment by comparing a variant without an FM (w/o FM) to those incorporating a generalist genomic model (Evo2 [7]) or an RNA-specific model (RNA-FM [9]). The results show that while the underlying $SO(2)$ geometric manifold provides a robust structural baseline, integrating FM alignment consistently improves metrics for Sequence and Secondary Structure Recovery. We attribute this to the co-evolutionary priors embedded in large language models, which provide sequence-level constraints that help guide the generation process to produce biologically plausible sequences. Furthermore, the model using domain-specific RNA-FM demonstrated slightly better but more consistent performance compared to the generalist Evo2. This suggests that, for

RNA inverse folding, evolutionary signals derived from RNA databases may provide more relevant semantic context than broad genomic representations.

Table 4: Ablation study on Foundation Model (FM) representation alignment.

Model Variant	Seq Recovery (↑)	SS Recovery (↑)	RMSD (Å, ↓)	
			RhoFold	AlphaFold3
GOLD (w/o FM)	0.6217	0.7792	11.23	12.58
GOLD (w/ Evo2)	0.6302	0.7922	10.85	12.40
GOLD (w/ RNA-FM)	**0.6380**	**0.8186**	**10.65**	**12.31**

Dual-Guided Sampling. To distinguish the effects of the guidance gradients from the external folding tools, we evaluate the SS Validity and Clash Score directly on the model's native outputs (Table 5). While the baseline model exhibits noticeable side-chain clashes in the absence of explicit guidance, the introduction of topological guidance (ω_{ss}) and spatial guidance (ω_{clash}) enhances Watson-Crick pairing validity and suppresses side-chain collisions. However, relying heavily on a single constraint (e.g., $\omega_{clash} = 1.0$) can distort the latent manifold, leading to a decrease in Sequence Recovery metric. By balancing both constraints ($\omega_{ss} = 1.0, \omega_{clash} = 0.1$), the dual-guided sampling setup achieves a favorable trade-off. It not only improves Sequence Recovery metric but also ensures the validity of the topology and minimizes spatial conflicts, indicating that incorporating these physical gradients can effectively guide the generation process without compromising overall sequence quality.

Table 5: Ablation study on the training-free Guidance mechanism.

Guidance Setup	Weights		Seq Recovery (↑)	SS Validity (↑)	Clash Score (↓)
	ω_{ss}	ω_{clash}			
Base (No Guidance)	0.0	0.0	0.6249	0.9040	4.51
SS Guidance Only	1.0	0.0	0.6267	0.9343	4.32
	2.0	0.0	0.6264	0.9356	4.43
	3.0	0.0	0.6268	**0.9382**	4.38
Clash Guidance Only	0.0	0.05	0.6362	0.9012	3.11
	0.0	0.1	0.6378	0.9028	2.76
	0.0	0.5	0.6353	0.9061	**2.28**
	0.0	1.0	0.6269	0.8938	2.77
Dual-Guidance	**1.0**	**0.1**	**0.6380**	0.9290	2.67

Additional appendices detail our evaluation results and mechanism analysis. Specifically, we evaluate the advantages of the unified latent manifold and compare our dynamic gradient guidance with direct feature utilization. We further assess the one-to-many nature of RNA inverse folding through sequence novelty, structural success rate, and intra-target diversity. Additionally, we investigate the interaction and sensitivity of the two physical guidance objectives through a two-dimensional weight sweep and KL-based analysis of the guided reverse process (Appendix G). Furthermore, we validate the effectiveness of the $SO(2)$-equivariant layer by comparing it with alternative sequence–torsion fusion strategies, including Trig-Concat and Fourier–FiLM (Appendix E), and we also analyze latent dimensionality impacts (Appendix F), and evaluate scalability on long RNA targets (Appendix D).

5.4 Case Studies

To qualitatively evaluate GOLD, we analyze the structural validity of the model's native outputs and use external tools to assess the actual foldability of the generated sequences.

Native Structural Validity. Figure 3 visualizes native local reconstructions without external relaxation. Although the unguided baseline exhibits significant atomic overlap, and the model using SS-only guidance improves base pairing but still exhibits residual steric conflicts (red spheres), dual guidance eliminates all collisions while maintaining pairing fidelity. This demonstrates that joint constraints are essential for local structural plausibility (more details see Appendix H).

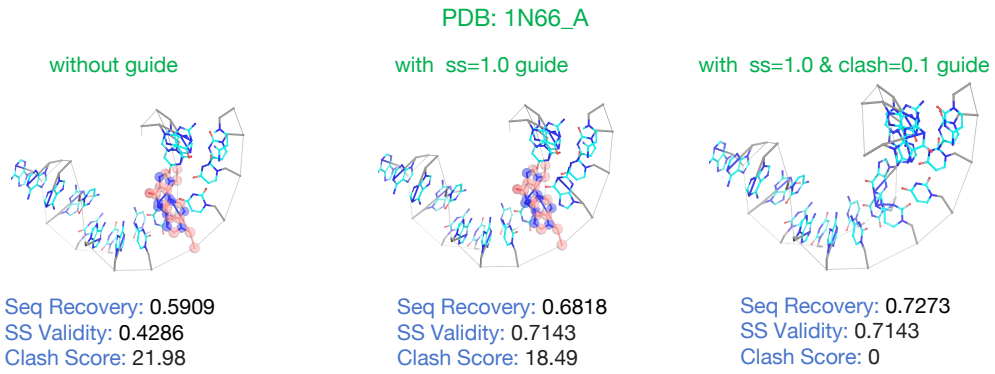

Figure 3: **Native structure reconstructions. Left**: Unguided (side-chain clashes). **Middle**: SS-only guided (improved pairing, but residual clashes). **Right**: Dual-guided (collisions resolved).

External Folding Validation. We assess foldability using AlphaFold3 predictions (Figure 4 for target **6PRV_D**). Baseline models (gRNAde, SOLD) exhibit spatial discrepancies compared to the actual structure, whereas the sequences designed by GOLD show a high degree of alignment with the native backbone. This confirms that our method can effectively translate geometry information into viable sequences (see more in Appendix I).

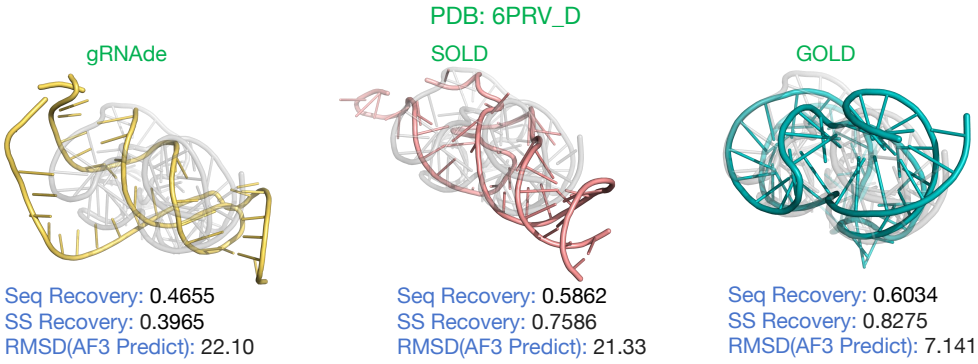

Figure 4: **AlphaFold3 folding validation for target** `6PRV_D`**.** Compared to baselines, the structure designed by GOLD aligns closely with the native ground truth (gray).

6 Conclusion

In this work, we propose GOLD, a geometry-aware latent diffusion framework for RNA inverse folding. By mapping discrete sequences and continuous side-chain geometries into a unified $SO(2)$-equivariant latent space, GOLD effectively addresses the modality gap in biomolecular representation. Furthermore, our dual-guided mechanism explicitly incorporates physical gradients into the sampling process. This strategy reduces steric clashes while enhancing the topological pairing validity without relying on complex reinforcement learning fine-tuning. Evaluations on internal benchmarks and CASP15 targets demonstrate GOLD exhibits robust structural fidelity and generalization capabilities.

Limitations and Future Work. First, calculating physical gradients during the sampling process increases inference time. Future work could explore distillation techniques to directly learn these physical rules, thereby achieving faster generation speeds. Second, while GOLD currently focuses on the standard fixed-backbone inverse folding paradigm, extending our latent manifold to support flexible-backbone co-design or multi-state conformational sampling remains a promising direction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 82394432, 92249302), the Shanghai Municipal Science and Technology Major Project (No. 2023SHZDZX02), and the AI for Science Program of the Shanghai Municipal Commission of Economy and Information (No. 2025-GZL-RGZN-BTBX-01008). We also acknowledge the support of the Novalnspire platform (Shanghai Academy of Artificial Intelligence for Science) and the CFFF platform (Fudan University) for providing computational resources.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 2024. doi: 10.1038/s41586-024-07487-w.

[2] Mirela Andronescu, Anthony P. Fejes, Frank Hutter, Holger H. Hoos, and Anne Condon. A new algorithm for rna secondary structure design. *Journal of Molecular Biology*, 2004. doi: 10.1016/j.jmb.2003.12.041.

[3] Minkyung Baek, Ryan McHugh, Ivan Anishchenko, Hanlun Jiang, David Baker, and Frank DiMaio. Accurate prediction of protein–nucleic acid complexes using rosettafoldna. *Nature Methods*, 2024. doi: 10.1038/s41592-023-02086-5.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. doi: 10.1109/TPAMI.2013.50.

[5] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=YCWjhGrJFD`.

[6] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016. doi: 10.18653/v1/K16-1002.

[7] Garyk Brixi, Matthew G. Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Jonathan C. Schmok, Ali Taghibakhshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K. Wang, Michael T. Pearce, Elana Simon, Etowah Adams, Zachary J. Amador, Euan A. Ashley, Stephen A. Baccus, Haoyu Dai, Steven Dillmann, Stefano Ermon, Daniel Guo, Michael H. Herschl, Rajesh Ilango, Ken Janik, Amy X. Lu, Reshma Mehta, Mohammad R. K. Mofrad, Madelena Y. Ng, Jaspreet Pannu, Christopher Ré, John St. John, Jeremy Sullivan, Joseph Tey, Ben Viggiano, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Sudarshan Pinglay, Dave P. Burke, Hani Goodarzi, Patrick D. Hsu, and Brian L. Hie. Genome modelling and design across all domains of life with evo 2. *Nature*, 2026. doi: 10.1038/s41586-026-10176-5.

[8] Anke Busch and Rolf Backofen. Info-rna—a fast approach to inverse rna folding. *Bioinformatics*, 2006. doi: 10.1093/bioinformatics/btl194.

[9] Jiayang Chen, Zhihang Hu, Siqi Sun, Qingxiong Tan, Yixuan Wang, Qinze Yu, Licheng Zong, Liang Hong, Jin Xiao, Tao Shen, et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022. URL https://doi.org/10.48550/arXiv.2204.00300.

[10] Ruizhe Chen, Dongyu Xue, Xiangxin Zhou, Zaixiang Zheng, xiangxiang Zeng, and Quanquan Gu. An all-atom generative model for designing protein complexes. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Afmi28vgIf.

[11] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

[12] Tulsi R. Damase, Roman Sukhovershin, Christian Boada, Francesca Taraballi, Roderic I. Pettigrew, and John P. Cooke. The limitless future of rna therapeutics. *Frontiers in Bioengineering and Biotechnology*, 2021. doi: 10.3389/fbioe.2021.628137.

[13] Rhiju Das and David Baker. Automated de novo prediction of native-like rna tertiary structures. *Proceedings of the National Academy of Sciences*, 104(37):14664–14669, 2007. doi: 10.1073/pnas.0703836104.

[14] Rhiju Das, Rachael C. Kretsch, Adam J. Simpkin, Thomas Mulvaney, Phillip Pham, Ramya Rangan, Fan Bu, Ronan M. Keegan, Maya Topf, Daniel J. Rigden, Zhichao Miao, and Eric Westhof. Assessment of three-dimensional rna structure prediction in casp15. *Proteins: Structure, Function, and Bioinformatics*, 2023. doi: 10.1002/prot.26602.

[15] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

[16] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=8OTPepXzeh.

[17] Zhangyang Gao, Cheng Tan, and Stan Z. Li. Pifold: Toward effective and efficient protein inverse folding. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oMsN9TYwJ0j.

[18] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. Rnaifold: a constraint programming algorithm for rna inverse folding and molecular design. *Journal of bioinformatics and computational biology*, 2013. doi: 10.1142/S0219720013500017.

[19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in neural information processing systems*, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[21] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of rna secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 1994. URL https://www.academia.edu/download/48689421/Fast_Folding_and_Comparison_of_RNA_Secon20160908-13624-1yg70az.pdf.

[22] Dongyue Hou, Shuai Zhang, Mengyao Ma, Hanbo Lin, Zheng Wan, Hui Zhao, Ruian Zhou, Xiao He, Xian Wei, Dianwen Ju, and Xian Zeng. A hyperbolic discrete diffusion 3d rna inverse folding model for functional rna design. *Journal of Chemical Information and Modeling*, 2025. doi: 10.1021/acs.jcim.5c00527.

[23] Tianmeng Hu, Yongzheng Cui, Biao Luo, and Ke Li. RIDER: 3d RNA inverse design with reinforcement learning-guided diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lDyS4Qg5Ww.

[24] Han Huang, Ziqian Lin, Dongchen He, Liang Hong, and Yu Li. Ribodiffusion: tertiary structure-based rna inverse folding with generative diffusion models. *Bioinformatics*, 2024. doi: 10.1093/bioinformatics/btae259.

[25] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf.

[26] John B. Ingraham, Max Baranov, Zak Costello, Karl W. Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M. Lord, Christopher Ng-Thow-Hing, Erik R. Van Vlack, Shan Tie, Vincent Xue, Sarah C. Cowles, Alan Leung, João V. Rodrigues, Claudio L. Morales-Perez, Alex M. Ayoub, Robin Green, Katherine Puentes, Frank Oplinger, Nishant V. Panwar, Fritz Obermeyer, Adam R. Root, Andrew L. Beam, Frank J. Poelwijk, and Gevorg Grigoryan. Illuminating protein space with a programmable generative model. *Nature*, 2023. doi: 10.1038/s41586-023-06728-8.

[27] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4.

[28] Chaitanya K. Joshi, Arian Rokkum Jamasb, Ramon Viñas Torné, Charles Harris, Simon V Mathis, Alex Morehead, Rishabh Anand, and Pietro Lio. gRNAde: Geometric deep learning for 3d rna inverse design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=lvw3UgeVxS.

[29] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://openreview.net/forum?id=33X9fd2-9FyZd.

[31] Neocles Leontis and Eric Westhof. Geometric nomenclature and classification of rna base pairs. *RNA*, 2001. doi: 10.1017/S1355838201002515.

[32] Xiner Li, Masatoshi Uehara, Xingyu Su, Gabriele Scalia, and Shuiwang Ji. A joint diffusion model with pre-trained priors for RNA sequence-structure co-design. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cpc63YrVWN.

[33] Sidney Lyayuga Lisanza, Jake Merle Gershon, Sam Tipps, Lucas Arnoldt, Samuel Hendel, Jeremiah Nelson Sims, Xinting Li, and David Baker. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. *bioRxiv*, 2023. URL https://www.biorxiv.org/content/10.1101/2023.05.08.539766v1.

[34] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7e3315fe390974fcf25e44a9445bd821-Paper.pdf.

[35] Runze Ma, Zhongyue Zhang, Zichen Wang, Will Hua, Jiahua Rao, Zhuomin Zhou, and Shuangjia Zheng. Riboflow: Conditional de novo RNA co-design via synergistic flow matching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=brV7U7svgS.

[36] Laura JW Murray, W Bryan Arendall III, David C Richardson, and Jane S Richardson. Rna backbone is rotameric. *Proceedings of the National Academy of Sciences*, 2003. doi: 10.1073/pnas.1835769100.

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/papers/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.pdf.

[38] Rafael Josip Penić, Tin Vlašić, Roland G Huber, Yue Wan, and Mile Šikić. Rinalmo: General-purpose rna language models can generalize well on structure prediction tasks. *Nature Communications*, 2025. doi: 10.1038/s41467-025-60872-5.

[39] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, 2018. doi: 0.1609/aaai.v32i1.11671.

[40] Vivek Ramanujan, Kushal Tirumala, Armen Aghajanyan, Luke Zettlemoyer, and Ali Farhadi. When worse is better: Navigating the compression generation trade-off in visual tokenization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=o8hWyJIgAV.

[41] Frederic Runge, Danny Stoll, Stefan Falkner, and Frank Hutter. Learning to design RNA. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ByfyHh05tQ.

[42] Frederic Runge, Karim Farid, Jörg KH Franke, and Frank Hutter. Rnabench: A comprehensive library for in silico rna modelling. *bioRxiv*, 2024. doi: 10.1101/2024.01.09.574794.

[43] Michael Sarver, Craig L Zirbel, Jesse Stombaugh, Ali Mokdad, and Neocles B Leontis. Fr3d: finding local and composite recurrent structural motifs in rna 3d structures. *Journal of mathematical biology*, 2008. doi: 10.1007/s00285-007-0110-x.

[44] Tao Shen, Zhihang Hu, Siqi Sun, Di Liu, Felix Wong, Jiuming Wang, Jiayang Chen, Yixuan Wang, Liang Hong, Jin Xiao, Liangzhen Zheng, Tejas Krishnamoorthi, Irwin King, Sheng Wang, Peng Yin, James J. Collins, and Yu Li. Accurate rna 3d structure prediction using a language model-based deep learning approach. *Nature Methods*, 2024. doi: 10.1038/s41592-024-02487-0.

[45] Qi Si, Xuyang Liu, Penglei Wang, Xin Guo, Yuan Qi, and Yuan Cheng. Structure-based rna design by step-wise optimization of latent diffusion model. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, 2026. doi: 10.1609/aaai.v40i2.37070.

[46] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

[47] 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*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

[48] Cheng Tan, Yijie Zhang, Zhangyang Gao, Bozhen Hu, Siyuan Li, Zicheng Liu, and Stan Z. Li. RDesign: Hierarchical data-efficient representation learning for tertiary structure-based RNA design. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RemfXx7ebP.

[49] Cheng Tan, Yijie Zhang, Zhangyang Gao, Hanqun Cao, Siyuan Li, Siqi Ma, Mathieu Blanchette, and Stan Z Li. R3design: deep tertiary structure-based rna sequence design and beyond. *Briefings in Bioinformatics*, 2025. doi: 10.1093/bib/bbae682.

[50] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. URL https://arxiv.org/abs/1802.08219.

[51] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Advances in neural information processing systems*, 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/5dca4c6b9e244d24a30b4c45601d9720-Paper.pdf.

[52] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

[53] Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Avantika Lal, Tommi Jaakkola, Sergey Levine, Aviv Regev, Hanchen, and Tommaso Biancalani. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=G328D1xt4W.

[54] Felix Wong, Dongchen He, Aarti Krishnan, Liang Hong, Alexander Z. Wang, Jiuming Wang, Zhihang Hu, Satotaka Omori, Alicia Li, Jiahua Rao, Qinze Yu, Wengong Jin, Tianqing Zhang, Katherine Ilia, Jack X. Chen, Shuangjia Zheng, Irwin King, Yu Li, and James J. Collins. Deep generative design of rna aptamers using structural predictions. *Nature Computational Science*, 2024. doi: 10.1038/s43588-024-00720-6.

[55] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/papers/Worrall_Harmonic_Networks_Deep_CVPR_2017_paper.pdf.

[56] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/yim23a.html.

[57] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[58] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

[59] Joseph N Zadeh, Conrad D Steenberg, Justin S Bois, Brian R Wolfe, Marshall B Pierce, Asif R Khan, Robert M Dirks, and Niles A Pierce. Nupack: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 2011. doi: 10.1002/jcc.21596.

[60] Min Zhao, Fan Bao, Chongxuan Li, and Jun Zhu. EGSDE: Unpaired image-to-image translation via energy-guided stochastic differential equations. In *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=xxgp42Qz6dL.

Appendix

Table of Contents

A Design Principles 17

B Algorithm Details 18

B.1 GeoVAE Details 18

B.2 Latent Diffusion: Score-Matching Training and Dual-Guided Sampling 19

B.3 Training Details 20

C Experimental Details 20

C.1 Dataset Construction 20

C.2 Baseline Implementations 22

C.3 Evaluation Metrics 23

D Performance and Cost Analysis Across Sequence Lengths 24

D.1 Performance of Long RNA 24

D.2 Inference Efficiency Across Varying Lengths 25

E Why SO(2) Layer is Necessary in GeoVAE? 26

E.1 Experimental Setup 26

E.2 Mitigating Modality Collapse 26

E.3 Structural Optimization Dynamics 27

E.4 Quantitative Comparison of Sequence–Torsion Fusion Strategies 27

F Discussion on Latent Dimension 28

F.1 Decoupling Reconstruction Fidelity from Generative Capacity 28

F.2 Robustness to Latent Dimension Scaling 28

G Extended Ablation Studies and Generative Evaluation 29

G.1 Unified Latent Manifold vs Raw-Space Co-Diffusion 29

G.2 Input Feature vs Gradient Guidance 31

G.3 One-to-Many Generative Evaluation 31

G.4 Interaction and Sensitivity of Physical Guidance 33

H Analysis of Dual-Guidance Dynamics 33

H.1 Topological Guidance (ω_{ss}) 34

H.2 Clash Guidance (ω_{clash}) 34

I Extended Case Studies 36

I.1 Case Study I 36

I.2 Case Study II 36

A Design Principles

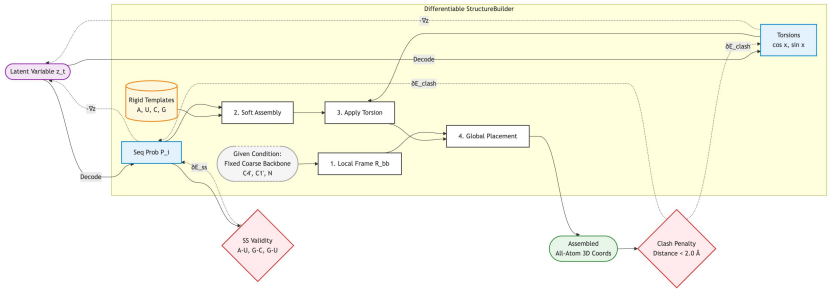

Figure 5: **StructureBuilder and Gradient Flow.** The architecture serves as a differentiable routing board, allowing gradients from 2D topological and 3D physical constraints to steer the latent sampling process.

Motivation: Enhancing Physical Validity via Geometric Modeling. Conventional RNA inverse folding models often lack a mechanism to bridge 1D sequences and 3D physical space. To address this issue, GOLD implements the differentiable **StructureBuilder** as its core geometric engine. This module realizes the operator $\mathcal{B}$ defined in main text Section 3. It utilizes sequence probabilities and side-chain angles to map latent predictions into 3D coordinates. This design serves a dual purpose. First, it couples discrete sequences with geometric angles at a physical level. This intrinsically anchors the joint generative distribution in a more physically plausible space. Second, it provides a fully differentiable pathway for direct physical energy feedback. This enables the model to explicitly resolve residual spatial conflicts.

Construction and Empirical Validation of the StructureBuilder. To map predicted latent variables back into physical space for spatial constraint perception, the StructureBuilder fuses sequence probabilities and side-chain angles with the fixed backbone via a four-stage differentiable cascade:

- **Stage I: Soft Semantic Assembly.** Weighted interpolation of standard base templates using the predicted sequence probabilities $\mathbf{P}_i$.
- **Stage II: Torsional Application.** Converting the predicted $\mathrm{SO}(2)$ angle χ_i into a 3D rotation matrix $\mathbf{R}_{\chi_i}$ to locally rotate the soft-assembled side-chain.
- **Stage III: Conditional Frame Construction.** Deterministically deriving local reference frames $\mathbf{R}_{bb}^{(i)}$ from the fixed backbone atoms.
- **Stage IV: Global Placement.** Projecting the side-chains into global coordinates $\mathbf{X}_{final}^{(i)}$, providing the 3D coordinate basis for calculating physical collision energies and gradients.

The physical foundation of this assembly mechanism relies on the planar rigid-body assumption of RNA nucleobases. To validate this assumption, we constructed a standard template bank $\mathbf{X}_{template} \in \mathbb{R}^{4\times14\times3}$. Specifically, we extracted the full-atom nucleotide coordinates from the GOLD Benchmark dataset. After aligning these extracted coordinates to a unified local relative coordinate system, we averaged them by base identity (A, U, C, G) to obtain idealized standard templates. Subsequently, we validated on the extensive GOLD Training set ($\approx$ 8,000 PDB structures). Restricting the internal DoF to the 1D rotation χ yields a **KAE** below 0.2Å. This result confirms that χ effectively captures side-chain geometry, which validates our strategy of using χ as the primary intermediate geometric variable.

Gradient Flow of the Physical Guidance Mechanism. The **StructureBuilder** acts as a bridge between physical signals and the sequence space during the sampling process (see Figure 5). At each timestep t, latent variables are decoded into sequence probabilities and side-chain angles, which are then assembled into coordinates. This supports two guidance pathways:

- **Pathway 1: Topological Guidance.** Calculating pairing probabilities to refine sequences.

- **Pathway 2: Physical Guidance.** Calculating clash energies between side-chains. The derived analytical gradients are decomposed to simultaneously update the side-chain orientations and sequence identities. Specifically:

  1. To the side-chain torsion angle: Adjusting orientations to avoid collisions.
  2. To the sequence probability: Modifying the base composition to reduce the probability of physical conflicts, resulting in more physically plausible sequences.

B Algorithm Details

B.1 GeoVAE Details

The GeoVAE is tasked with the objective of compressing discrete 1D nucleic acid sequences and continuous 3D spatial geometries into a smooth, compact, low-dimensional continuous manifold. To ensure that the latent space possesses both "geometric optimal" and "biological semantics", we incorporated SO(2) Rotation-Aware Layers into the forward pass to process torsion angles, and introduced a "Foundation Model Representation Alignment" mechanism to distill large-scale pre-trained semantic information (e.g., from RNA-FM) into the latent space.

The complete training procedure is detailed in Algorithm 1. During the encoding phase, the model outputs the posterior distribution parameters μ and $\log \sigma^2$. Concurrently, it projects the externally provided LLM sequence embeddings H_{LLM} into the latent space (yielding $z_{teacher}$), and employs an alignment loss ($\mathcal{L}_{repa}$) to constrain the VAE's distribution center μ to a desired manifold. During the decoding phase, the model first reconstructs the sequence logits and continuous angles. Subsequently, by utilizing the operator $\mathcal{B}$, it recovers the spatial coordinates $\hat{X}_{sc}$ of side-chain conditioned on the coarse backbone X_{bb} while maintaining the connectivity of computational graph. The multi-task loss function is scaled by a set of hyperparameters and optimized jointly end-to-end.

Algorithm 1 GeoVAE Training Forward Pass and Loss Computation

Require: Graph batch $\mathcal{G} = \{S, \chi, X_{bb}, X_{sc}\}$, LLM embeddings H_{LLM}, Loss weights $\mathbf{w}$
Ensure: Updated Encoder (ϕ), Decoder (θ), and Projection (ψ) parameters

// 1. SO(2)-Aware Encoding Phase
$E_{seq} \leftarrow \text{Embedding}(S)$
$H_{enc} \leftarrow \text{RotationAwareEncode}_{\phi}(E_{seq}, \chi)$ ▷ *Fuses seq and angles in $SO(2)$ domain*
$\mu, \log \sigma^2 \leftarrow \text{GraphEncoder}_{\phi}(H_{enc}, X_{coarse})$
// 2. Latent Reparameterization & LLM Alignment
$\epsilon \sim \mathcal{N}(0, I)$
$z \leftarrow \mu + \epsilon \odot \exp(\frac{1}{2} \log \sigma^2)$
$z_{teacher} \leftarrow \text{LatentProjection}_{\psi}(H_{LLM})$ ▷ *Distill RNA-FM semantics*
$\mathcal{L}_{repa} \leftarrow 1 - \cos(\mu, z_{teacher})$
// 3. Decoding Phase
$H_{dec} \leftarrow \text{GraphDecoder}_{\theta}(z)$
$\hat{S}_{logits}, \hat{\chi} \leftarrow \text{RotationAwareDecode}_{\theta}(H_{dec})$
// 4. Differentiable Structure Building
$\hat{P}_{seq} \leftarrow \text{Softmax}(\hat{S}_{logits})$ ▷ *Differentiable relaxation*
$\hat{X}_{sc} \leftarrow \text{StructureBuilder}(\hat{P}_{seq}, \hat{\chi}, X_{bb})$ ▷ *Recovers 3D coordinates*
// 5. Multi-task Loss Computation
$\mathcal{L}_{seq} \leftarrow \text{CrossEntropy}(\hat{S}_{logits}, S)$
$\mathcal{L}_{angle} \leftarrow \text{MSE}(\hat{\chi}, \chi)$
$\mathcal{L}_{coord} \leftarrow \text{MSE}(\hat{X}_{sc}, X_{sc})$
$\mathcal{L}_{KL} \leftarrow -\frac{1}{2} \sum(1 + \log \sigma^2 - \mu^2 - \exp(\log \sigma^2))$
$\mathcal{L}_{total} \leftarrow w_{seq}\mathcal{L}_{seq} + w_{angle}\mathcal{L}_{angle} + w_{coord}\mathcal{L}_{coord} + w_{kl}\mathcal{L}_{KL} + w_{repa}\mathcal{L}_{repa}$
Update ϕ, θ, ψ by descending $\nabla_{\phi,\theta,\psi}\mathcal{L}_{total}$

B.2 Latent Diffusion: Score-Matching Training and Dual-Guided Sampling

Our diffusion generative framework is composed of two core operational phases: continuous score-matching on the latent manifold (Training) and physics-aware reverse denoising (Inference).

Diffusion Training Phase. As detailed in Algorithm 2, we freeze the weights of the pre-trained GeoVAE, utilizing it as a high-fidelity latent tokenizer during training. For a given native macromolecular graph, we extract its latent representation z_0 through the GeoVAE encoder. We uniformly sample a timestep $t \sim \mathcal{U}(1, T)$ and inject Gaussian noise into z_0 according to a pre-defined cosine noise schedule, yielding the noisy state z_t. Our denoising backbone (DiT) is designed to directly predict the expectation of the clean latent variable $\hat{z}_0$ and the network parameters θ are optimized by minimizing the loss of MSE.

Algorithm 2 Latent Diffusion Training

Require: Training graph batch $\mathcal{G} = \{S, \chi, X_{bb}\}$, Frozen GeoVAE Encoder (ϕ^*), DiT model (θ)
Ensure: Updated DiT parameters θ

// 1. Extract Clean Latent Representation
$\mu, \log \sigma^2 \leftarrow \phi^*(S, \chi, X_{bb})$
$\epsilon_{reparameter} \sim \mathcal{N}(0, I)$
$z_0 \leftarrow \mu + \epsilon_{reparameter} \odot \exp(\frac{1}{2} \log \sigma^2)$ ▷ *Ground-truth clean latent*
// 2. Forward Diffusion (Add Noise)
$t \sim \mathcal{U}(1, T)$ ▷ *Sample random timestep*
$\epsilon_{noise} \sim \mathcal{N}(0, I)$
$\bar{\alpha}_t \leftarrow \text{CosineSchedule}(t)$
$z_t \leftarrow \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_{noise}$ ▷ *Obtain noisy latent state*
// 3. Denoising Prediction and Optimization
$\hat{z}_0 \leftarrow \text{DiT}_\theta(z_t, t, X_{bb})$ ▷ *Model predicts clean z_0*
$\mathcal{L}_{latent} \leftarrow \text{MSE}(\hat{z}_0, z_0)$ ▷ *Score-matching objective*
Update θ by descending $\nabla_\theta \mathcal{L}_{latent}$

Dual-Guided Sampling Phase. During inference, we propose a dual-guided sampling algorithm (Algorithm 3). At each denoise timestep t, we leverage the frozen GeoVAE decoder to project the predicted $\hat{z}_0$ back into the 1D/3D biological space. This allows us to compute the topological pairing energy $\mathcal{E}_{ss}$ and spatial repulsion energy $\mathcal{E}_{clash}$ natively in the physical space. The aggregated energy gradients are then backpropagated to the denoised latent state z_t. To prevent excessive guidance from disrupting the smoothness of the manifold, we dynamically anneal the guidance scale ($\gamma_t = t/T$), imposing strong constraints in the early stages while maintaining micro-fidelity in the final stages.

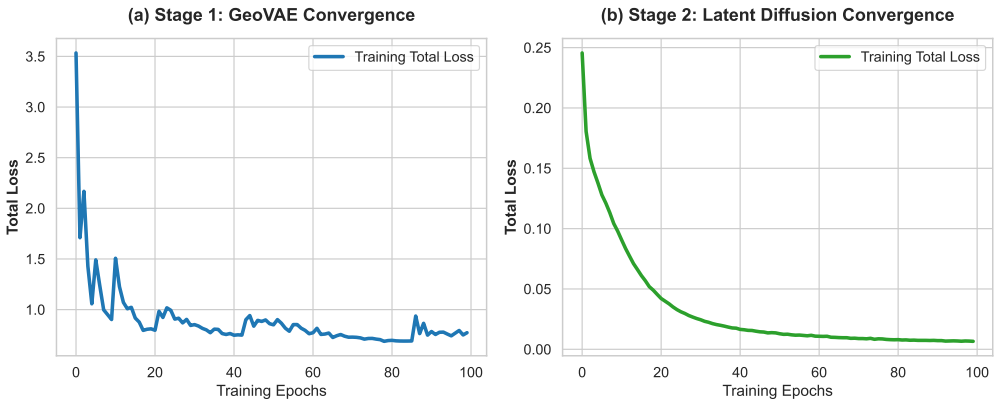

Figure 6: **Convergence dynamics of the two-stage training process.** (a) Total validation loss of GeoVAE. (b) Total training loss of the Latent Diffusion stage.

Algorithm 3 Dual-Guided Latent Diffusion Sampling

Require: Coarse backbone X_{bb}, Native SS graph E_{ss}, Total steps T, Guidance weights $\omega_{ss}, \omega_{clash}$
Ensure: Sampled sequence $\hat{S}$ and fine-grained side-chain coordinates $\hat{X}_{sc}$

$z_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T$ **to** 1 **do**
  $\hat{z}_0 \leftarrow \text{DiT}_\theta(z_t, t, X_{bb})$ ▷ *Predict clean latent expectation*
  if $\omega_{ss} > 0$ **or** $\omega_{clash} > 0$ **then**
    Require gradient tracking for z_t
    // 1. Cross-space Differentiable Projection
    $\hat{S}_{logits}, \hat{\chi}, \hat{X}_{sc} \leftarrow \text{GeoVAE_Decoder}_{\theta^*}(\hat{z}_0, X_{bb})$
    $\hat{P}_{seq} \leftarrow \text{Softmax}(\hat{S}_{logits})$
    // 2. Physical Energy Computation
    $\mathcal{E}_{ss} \leftarrow \text{ComputePairingEnergy}(\hat{P}_{seq}, E_{ss})$ ▷ *Evaluates Watson-Crick rules*
    $\mathcal{E}_{clash} \leftarrow \text{ComputeClashEnergy}(\hat{X}_{sc}, \text{threshold} = 2.0\text{Å})$
    $\mathcal{E}_{total} \leftarrow \omega_{ss}\mathcal{E}_{ss} + \omega_{clash}\mathcal{E}_{clash}$
    // 3. Backpropagation and Dynamic Annealed Update
    $g \leftarrow \nabla_{z_t}\mathcal{E}_{total}$ ▷ *Backpropagate energy gradient to latent*
    $\gamma_t \leftarrow t/T$ ▷ *Dynamic linear annealing scale*
    $z_t \leftarrow z_t - \gamma_t \cdot g$ ▷ *Apply physics-based guidance*
  end if
  $z_{t-1} \leftarrow \text{Denoise_Step}(z_t, \hat{z}_0, t)$ ▷ *Standard Denoised manifold transition*
end for
// Final Decode
$\hat{S}_{logits}, \hat{A}, \hat{X}_{sc} \leftarrow \text{GeoVAE_Decoder}_{\theta^*}(z_0, X_{bb})$
Return $\arg\max(\hat{S}_{logits})$, $\hat{X}_{sc}$

B.3 Training Details

To ensure reproducibility, we detail the hyperparameter configurations for our two-stage architecture in Table 6.

Stage 1: GeoVAE Training. The GeoVAE was trained on a single NVIDIA A100 (80GB) GPU, with each epoch requiring approximately 30 minutes. We utilized the AdamW optimizer with a base learning rate of 5×10^{-4} for 100 epochs. To prioritize 3D backbone reconstruction, we used asymmetric loss weights: coordinates (10.0), angles (5.0), sequence (1.0), and KL (0.1). As illustrated in Figure 6 (a), the total validation loss decays rapidly during the initial phase and reaches a stable plateau around 80 epochs.

Stage 2: Latent Diffusion Training. Following the latent manifold, the diffusion network was trained on the latent space. Under the same hardware environment, each epoch of the Latent Diffusion stage required approximately 1 hour. The model was trained for 100 epochs with a learning rate of 1×10^{-4} and a batch size of 32. As shown in Figure 6 (b), the overall training loss exhibits a smooth decay, confirming the stability of the diffusion process within the compressed latent space.

Computational Cost and Inference Latency. Using the latent compression strategy, GOLD avoids the computational explosion commonly found in all-atom diffusion models. The entire pipeline achieves convergence within roughly 100 A100-hours. During inference, generating a single RNA 3D structure takes only a few seconds (please see details in appendix D.2), demonstrating potential for high-throughput RNA design tasks.

C Experimental Details

C.1 Dataset Construction

GOLD Benchmark: To train the GOLD and conduct inverse folding evaluations, we constructed the high-quality GOLD Benchmark. Unlike studies that simply use raw PDB files directly, we designed an optimized workflow to obtain high-quality data. The detailed procedures are as follows:

Table 6: Detailed architectural configurations and training hyperparameters for the two-stage generative framework (GeoVAE and Latent Diffusion).

Training Stage	Module	Hyperparameter	Value
Stage 1: GeoVAE	*Architecture*	Node Hidden Dimension (d_{model})	256
		Latent Dimension (d_{latent})	8
		Encoder/Decoder Layers	4
		Attention Heads	4
		Dropout Rate	0.1
		Pre-trained LLM Embed Dimension (RNA-FM)	640
	Optimization	Base Learning Rate	5×10^{-4}
		Weight Decay	1×10^{-4}
		Batch Size (Per GPU)	1
		Gradient Accumulation Steps	16
		Training Epochs	100
		Gradient Clipping	1.0
		Loss Weights ($w_{coord}, w_{angle}, w_{seq}, w_{kl}$)	10.0, 5.0, 1.0, 0.1
Stage 2: Latent Diffusion	*GVP Condition*	Node Hidden Dim (Scalar, Vector)	(128, 16)
		Edge Hidden Dim (Scalar, Vector)	(32, 1)
		GVP Layers	4
		Spatial k-NN Graph	$k = 10$
	DiT Backbone	Transformer Layers	12
		Encoder Embedding Dimension	128
		FFN Embedding Dimension	512
		Attention Heads	4
	Diffusion & Optim	Diffusion Timesteps	100
		Noise Schedule	Cosine
		Base Learning Rate	1×10^{-4}
		Weight Decay	1×10^{-4}
		Batch Size	32
		Training Epochs	100
		Gradient Accumulation Steps	1

1. **Initial Collection and Filtering:** We aggregated all publicly available 3D structures containing RNA from the Protein Data Bank (PDB) released up to August 19, 2025. Initially, we excluded short fragments with a sequence length of less than 5 nt, as well as low-quality structures with a resolution worse than 4.0 Å.

2. **RNA Entity Definition and Topology Extraction:** For downstream processing, we defined fundamental RNA entities by clustering RNA chains according to inter-chain base-pairing interactions identified by FR3D [43]. Specifically, we generated an RNA-chain interaction graph where an edge between two chains was established if they shared at least three inter-chain base pairs, and this inter-chain base-pair count amounted to no less than 15% of the shorter chain's length. Groups of connected RNA chains in this graph were merged into single integrated RNA entities, whereas isolated chains lacking such links were treated as standalone entities. The ground-truth secondary structures (gt ss) were derived simultaneously from this FR3D pairing network. These RNA entities served as the fundamental units for all subsequent preprocessing procedures.

3. **Multi-modal Consistency Check:** Since raw PDB files frequently contain missing residues or mismatched annotations, we re-parsed the crystalline coordinates to extract physically present nucleotides (`seq`), which were aligned against the ground-truth reference sequence. Furthermore, for the RNA entities defined above, we enforced consistency validation between the 1D sequence, 2D topology and 3D coordinates. Any sample exhibiting mismatches in length or composition in these modalities was classified as noise and filtered out.

4. **Chemical Modification Normalization:** Native nucleic acids are rich in complex chemical modifications. To align with the standard 4-letter discrete vocabulary (A, U, C, G) required by foundation models and our GeoVAE, we implemented a precise mapping dictionary to revert non-standard modified residues to their canonical counterparts.

5. **Dual-level Homology Isolation and Splitting:** To definitively prevent data leakage caused by homologous sequences across the evaluation sets, we applied rigorous clustering across two independent representation spaces. This prevents highly similar samples from being distributed across different data splits (train, validation, or test):

- **Sequence-level:** We utilized MMseqs2 for sequence clustering, setting the sequence identity threshold to a stringent 0.3 (30%).
- **Structural-level:** We further utilized US-align to calculate 3D topological alignment scores, clustering structures with a similarity (TM-score) exceeding 0.45.

Any samples with the sequence identity ($> 30\%$) or structural similarity (TM-score > 0.45) exceeding thresholds were assigned to the same cluster. During dataset partitioning, each cluster was treated as an indivisible unit and assigned entirely to a specific split. This strategy eliminates homologous information leakage between the training and evaluation sets, ensuring that the evaluation of generative models is reliable.

The non-redundant core dataset selected through this rigorous pipeline was split into training, validation, and testing sets (7583/1144/598). Importantly, to ensure fair benchmarking, the extracted ground-truth secondary structure (ss) information was strictly excluded from the training phase of all evaluated algorithms. Please see details in Appendix G.2. This topology information was exclusively reserved for downstream evaluation metrics and zero-shot inference guidance.

CASP15 Targets: To evaluate the out-of-distribution (OOD) generalization capability on novel structures, we further compiled an independent test set from the CASP15 challenge. This challenging set comprises 60 sequence-structure pairs, providing a benchmark for assessing the model's robustness on unseen landscapes.

C.2 Baseline Implementations

To ensure a fair comparison, all baseline models were trained and evaluated on our standardized GOLD Benchmark dataset splits. We adapted the models originally designed for proteins to accommodate RNA structures. The specific implementation details are as follows:

PiFold: PiFold was originally proposed as a highly efficient graph neural network architecture for protein structure design [17]. To fairly evaluate this model on RNA inverse folding task, we performed specific domain adaptation based on its official open-source implementation (`https://github.com/A4Bio/PiFold`) to accommodate nucleic acids. The specific modifications are as follows:

- **Atomic Representation Substitution:** We replaced the amino acid backbone atoms (N, CA, C, O) relied upon for spatial k-NN graph construction with key atoms representing the coarse-grained RNA backbone (namely, the C1' atom of the phosphate backbone, the C4' atom of the ribose ring, and the N1/N9 pseudo-atoms representing the spatial orientation of the nucleobases).
- **Feature:** At the feature engineering level, we retained PiFold's core "Virtual Center" node representation mechanism, as well as the computational logic for multi-spatial edge features, which comprise multi-dimensional pairwise distances, dihedral angles, and spatial orientations.
- **Network and Output Adaptation:** The model backbone remains composed of multiple layers of `ProDesign_Encoder` and `ProDesign_Decoder`, aggregating local geometric information through graph message passing mechanism. At the output layer, we modified the classification head to predict the probability distribution over the four standard RNA nucleotides (A, U, C, G) instead of the 20 canonical amino acids.

During the training phase, we utilized the exact same training and validation splits as our GOLD benchmark to ensure the comparability of the results, and optimized the parameters end-to-end using the standard CrossEntropyLoss.

RDesign: RDesign is a native graph neural network architecture explicitly developed for RNA 3D structure inverse folding [48]. To ensure a fair evaluation on our GOLD Benchmark, we seamlessly integrated its official open-source implementation (`https://github.com/A4Bio/RDesign`) into our standardized training and evaluation pipeline.

RDesign constructs spatial k-NN graphs using coarse-grained atoms. We configured the model with a hidden dimension of 128, along with 3 encoder and 3 decoder layers. It predicts the categorical probability distribution over the four nucleotides. For a rigorous comparison, RDesign was trained on

our GOLD dataset splits using the AdamW optimizer (with a base learning rate of 3×10^{-4}) with a batch size of 8 for 100 epochs, minimizing the standard CrossEntropyLoss.

gRNAde: gRNAde is an autoregressive RNA inverse folding model utilizing Geometric Vector Perceptrons (GVPs) [28]. Based on its open-source implementation (`https://github.com/chaitjo/geometric-rna-design`), we constructed spatial graphs using coarse-grained RNA atomic coordinates (capped at a 4.5 Å radius and 32 nearest neighbors), extracting multi-dimensional scalar and vector features (including 32-bin RBF and 32-dim positional encodings). We configured the model with 4 GVP convolutional layers, node hidden dimensions of 128 scalar and 16 vector, edge hidden dimensions of 64 scalar and 4 vector. For a rigorous comparison, gRNAde was trained on our GOLD dataset splits. We optimized the model using the Adam optimizer (with a base learning rate of 1×10^{-4}) and a label smoothing factor of 0.05, employing dynamic batch sampling (maximum 3000 nodes per batch) for 100 epochs.

RiboDiffusion: RiboDiffusion is a continuous-time diffusion model designed for RNA inverse folding [24]. Since its official open-source repository (`https://github.com/ml4bio/RiboDiffusion`) lacks training scripts, we implemented a custom training pipeline from scratch to enable training on our GOLD dataset splits. Architecturally, the model constructs spatial k-NN graphs ($k = 10$) and extracts 3D geometric features via 4 Geometric Vector Perceptron (GVP) convolutional layers (node hidden dimensions configured to 512 scalar and 128 vector). These representations are subsequently fed into an 8-layer Transformer equipped with 16 attention heads for sequence denoising. We utilized a cosine noise schedule and optimized the network using the AdamW optimizer (with a base learning rate of 1×10^{-4} and a weight decay of 0.001) and a batch size of 1 with 8 gradient accumulation steps over 1000 training epochs.

SOLD: SOLD is a latent diffusion network that leverages sequence foundation models for inverse folding [45]. Based on its official open-source repository (`https://github.com/darkflash03/SOLD`), we reproduced its two-stage training pipeline on our GOLD dataset splits. In the first stage (feature compression), an MLP autoencoder compresses 640-dimensional RNA-FM pre-trained sequence embeddings into a 32-dimensional latent space, paired with a 3-layer MLP decoder (dropout 0.1) to predict nucleotide probabilities. This stage was trained for 100 epochs using the Adam optimizer (base learning rate 1×10^{-4}) with a batch size of 1 and 16 gradient accumulation steps. In the second stage (latent diffusion), the model executes a continuous-time diffusion process over 100 denoising steps. The denoising architecture combines 4 GVP convolutional layers (node hidden dimensions configured to 512 scalar and 128 vector) and an 8-layer Transformer equipped with 16 attention heads. This diffusion stage was trained end-to-end for 100 epochs using the AdamW optimizer (learning rate 1×10^{-4}, weight decay 0.01) with a batch size of 64.

C.3 Evaluation Metrics

To comprehensively evaluate the performance of generative models in sequence, topology, and spatial dimensions, we formalized the mathematical definitions of all evaluation metrics in our pipeline. Notably, for secondary structure evaluation, we engineered two independent computational protocols tailored to different experimental objectives.

Sequence Recovery (Seq Recovery): Measures the proportion of exact character matches between the generated sequence $\hat{S}$ and the ground truth sequence S^{gt} at each aligned position over length L:

$$\text{Seq Recovery} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{I}(\hat{s}_i = s_i^{gt}) \tag{11}$$

Secondary Structure Recovery (SS Recovery): In the **Main Results** section, to fairly evaluate the intrinsic topological potential of the generated sequences while eliminating the inherent prediction bias of external tools, we input both the generated sequence $\hat{S}$ and the ground truth sequence S^{gt} into the ViennaRNA folding engine. We then compute the character-level match between their full-length dot-bracket strings:

$$\text{SS Recovery} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{I}\left(\text{ViennaRNA}(\hat{S})_i = \text{ViennaRNA}(S_{gt})_i\right) \tag{12}$$

RMSD based on RhoFold & AlphaFold3: To evaluate the 3D physical foldability of the generated sequences, we employed RhoFold and AlphaFold3 (AF3) as folding oracles. For RhoFold, we executed the official pretrained weights in `single_seq_pred` mode locally. For RNA structure prediction using AF3, five predictions were generated using a fixed random seed (seed = 2026), and only the top-ranked poses selected by ranking score. All other parameters remained at their default values.

Given that sequence-based alignment fails in inverse folding due to sequence mutations, we developed a global RMSD algorithm based on the backbone C4' atoms that considers only the topological structure. Given the C4' coordinates of the ground truth structure $X \in \mathbb{R}^{L\times 3}$ and the predicted structure $Y \in \mathbb{R}^{L\times 3}$, we map the residues by position and align them to the origin ($\bar{X} = X - \mu_X, \bar{Y} = Y - \mu_Y$). The covariance matrix $H = \bar{Y}^T\bar{X}$ undergoes Singular Value Decomposition (SVD) to analytically solve for the optimal Kabsch rotation matrix [29] R:

$$R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{pmatrix} U^T \tag{13}$$

The RMSD is then calculated:

$$\text{RMSD} = \sqrt{\frac{1}{L}\sum_{i=1}^{L} \left\|\bar{X}_i - (\bar{Y}_i R)\right\|^2} \tag{14}$$

SS Validity in Ablation Analysis: Unlike our main experiments that rely on external secondary structure predictors, analyzing the ablation of the topological guidance parameter (ω_{ss}) requires a tool-agnostic perspective to isolate its direct impact. To avoid any biases introduced by external prediction tools, we propose a deterministic physical verification metric. Based on the ground-truth base-pairing graph E_{ss} extracted directly from the native 3D structure, this metric strictly verifies whether the generated sequence successfully satisfies Watson-Crick (A-U, G-C) or Wobble (G-U) base-pairing rules at the designated topological edges:

$$\text{SS Validity} = \frac{1}{|E_{ss}|}\sum_{(i,j)\in E_{ss}} \mathbb{I}\left((\hat{s}_i, \hat{s}_j) \in \mathcal{P}_{valid}\right) \tag{15}$$

where $\mathcal{P}_{valid} = \{(\text{A,U}), (\text{U,A}), (\text{G,C}), (\text{C,G}), (\text{G,U}), (\text{U,G})\}$. This direct metric provides an clear perspective for evaluating the effectiveness of topological guidance.

Clash Score in Ablation Analysis: Standard structural evaluation pipelines often employ coordinate relaxation tools, which automatically resolve atomic clashes post-hoc. Relying on such post-processing would inevitably obscure the direct contribution of our spatial energy guidance. To independently analyze the inherent capabilities of our guidance mechanism in ablation studies, we directly calculate the Clash Score on the unrelaxed 3D coordinates generated by our model. This metric is defined as the number of non-bonded atomic pairs with a spatial distance below $2.0\,\text{Å}$ per 1000 atoms. To avoid falsely penalizing natural covalent bonds, we strictly mask the distances between adjacent residues (i.e., $|i-j| \leq 1$):

$$\text{Clash Score} = \frac{1000}{N_{atoms}}\sum_{i,j:|i-j|>1} \mathbb{I}(d(i,j) < 2.0\,\text{Å}) \tag{16}$$

D Performance and Cost Analysis on Sequence Lengths

In RNA 3D inverse folding, the solution space for geometric modeling and sequence generation expands rapidly as sequence length increases. To evaluate the model's performance and computational scalability on extended sequences, we conducted an analysis from two aspects: generation quality and inference cost.

D.1 Performance across RNA of different lengths.

To compare generation quality, we partitioned the GOLD Benchmark test set into short-to-medium sequences ($L \leq 100$) and long sequences ($L > 100$). As shown in Table 7, the transition to the

$L > 100$ regime poses an extreme challenge for the entire field, evidenced by a significant degradation in RMSD across all baseline models. However, under these challenging conditions, GOLD exhibits the strongest relative robustness. While maintaining leading structure metrics, GOLD successfully achieves high Sequence Recovery and outperforms the strongest baseline (SOLD) in 2D topological pairing accuracy.

Table 7: Performance breakdown across different sequence length regimes on the GOLD Benchmark.

Method	**Seq Recovery (↑)**	**SS Recovery (↑)**	**3D RMSD (Å, ↓)**	
			RhoFold	**AlphaFold3**
Short-to-Medium Sequences ($L \leq 100$)				
PiFold	0.4237	0.6289	15.81	17.36
RDesign	0.4443	0.6918	13.06	13.47
gRNAde	0.5440	0.6614	13.75	14.38
RiboDiffusion	0.5158	0.7061	11.56	12.41
SOLD	0.5837	0.7479	11.07	12.37
GOLD (Dual-Guided)	**0.6397**	**0.8332**	**8.968**	**10.80**
Long Sequences ($L > 100$)				
PiFold	0.3776	0.4877	38.37	37.19
RDesign	0.4245	0.4554	36.15	37.39
gRNAde	0.5609	0.4302	37.46	39.33
RiboDiffusion	0.4684	0.5445	34.41	32.95
SOLD	0.5839	0.5868	34.49	33.01
GOLD (Dual-Guided)	**0.6289**	**0.6237**	**33.43**	**32.67**

D.2 Inference Efficiency Across Different Lengths.

To evaluate the scalability of our approach in practical scenarios, we measure the generation latency and peak GPU memory consumption across four RNA targets with lengths (L) ranging from 44 to 308. All tests are conducted in a unified hardware environment using a single NVIDIA A100 (80GB) GPU.

As detailed in Table 8, compressing the generation process into a low-dimensional latent manifold effectively avoids the heavy memory bottlenecks typically associated with explicit spatial coordinate generation. Consequently, **GOLD (Base)** demonstrates highly competitive computational efficiency: it achieves comparable inference cost to the unimodal baseline SOLD, while maintaining a peak GPU of only around 2 GB even for sequences longer than 300 nucleotides.

Furthermore, applying the dual-guided mechanism introduces negligible GPU overhead across all lengths. In terms of inference speed, calculating the physical constraint gradients at each denoising step incurs only a minimal time penalty for shorter RNA sequences. Although this inference cost inevitably scales for longer targets, the absolute time remains well within practical limits for downstream RNA design tasks. In summary, these results demonstrate that GOLD provides a scalable architecture that effectively balances structural generation quality with computational efficiency.

Table 8: Inference cost across RNA targets. Time (T) in seconds, Peak Memory (M) in MB.

PDB ID	**Len**	**Inference Time (s) ↓**			**Peak GPU Mem (MB) ↓**		
		SOLD	**GOLD$_{Base}$**	**GOLD$_{Dual}$**	**SOLD**	**GOLD$_{Base}$**	**GOLD$_{Dual}$**
1A60_A	44	5.12	5.37	7.36	1486	1525	1529
8QO4_A	156	5.34	5.93	9.69	1598	1693	1708
5TC1_R	242	6.28	6.85	11.61	1786	1811	1847
6N7X_R	308	7.05	7.61	14.78	1876	2043	2088

E Why SO(2) Layer is Necessary in GeoVAE ?

In Section 4.2 of the main text, we introduced a $SO(2)$-based rotation-aware architecture for the GeoVAE encoder and decoder. In this section, we evaluate the advantages of this design through comparisons with alternative sequence–torsion fusion strategies. We first analyze feature-level sensitivity and training dynamics against a traditional concatenation baseline, followed by a quantitative comparison with a stronger Fourier–FiLM baseline.

E.1 Experimental Setup

To isolate the effects of the fusion mechanism, we trained two models under identical settings for the perturbation and training-dynamics analyses: our $SO(2)$ **GeoVAE** and a baseline model that directly concatenates sequence embeddings and trigonometric angle vectors (**Trig-Concat Baseline**). Both models were supervised exclusively by sequence and angle reconstruction losses. The subsequent quantitative fusion comparison adopts the complete GeoVAE training objective, including coordinate reconstruction supervision.

Evaluation Metric. To evaluate the structural reconstruction accuracy, we employ the Kinematic Approximation Error (**KAE**) in Section 3. This metric provides a direct geometric assessment of the decoded features.

E.2 Mitigating Modality Collapse

To probe how each architecture processes multi-modal features internally, we conducted an input-level spatial perturbation test. We selected a target RNA nucleotide from the GOLD Benchmark test set, systematically perturbed its input torsion angle from $-\pi$ to π, and recorded the corresponding changes in the predicted sequence probability ($\hat{P}_{seq}$).

As illustrated in Figure 7 (left), the sequence probability predicted by the Trig-Concat baseline remains largely invariant regardless of how the input angle is manipulated. This observation suggests that the Trig-Concat model exhibits a form of modal collapse, likely due to an optimization imbalance between discrete sequence classification and continuous angle regression. To rapidly minimize the total loss early in training, the network relies primarily on the discrete sequence features, thereby ignoring the continuous spatial inputs.

Conversely, the $SO(2)$ model demonstrates consistent responsiveness to spatial perturbations (Figure 7, right). In the $SO(2)$ architecture, sequence (magnitude) and angles (phase) are explicitly coupled through complex multiplication. Altering the input angle inherently shifts the spatial context processed by the subsequent Transformer layers, predictably modifying the sequence output. This structural coupling prevents the network from bypassing geometric cues, thereby encouraging a more reliable integration of sequence and spatial constraints.

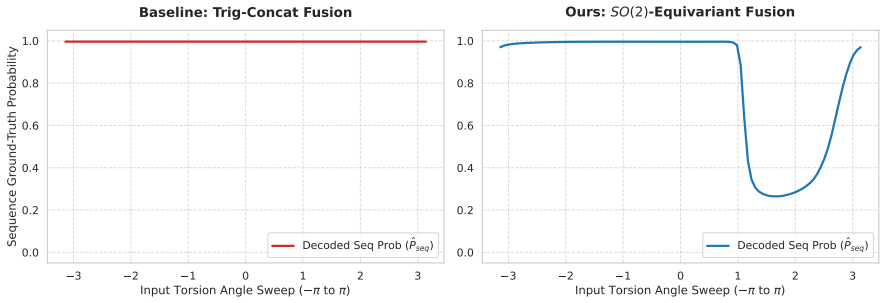

Figure 7: **modality collapse vs. synergistic coupling.** Comparison of predicted sequence probability ($\hat{P}_{seq}$) in response to a continuous input-level angular sweep. The Trig-Concat Baseline (left) exhibits an entirely flat response, indicating that the network zeroed out spatial weights early in training. In contrast, the $SO(2)$ GeoVAE (right) demonstrates sensitivity to geometric perturbations, proving that its architecture prevents information truncation and forces the synergistic coupling of sequence and geometry.

E.3 Structural Optimization Dynamics

These differences in feature processing directly affect the overall training dynamics of the two models. As shown in Figure 8, there are significant differences in their structural optimization approaches. Because the Trig-Concat baseline relies heavily on discrete sequence features and largely ignores continuous spatial inputs during the early stages of training (as shown by the perturbation test), its initial structural predictions lack geometric guidance. Consequently, its KAE gets stuck in an optimization plateau near 1.0 Å for over 20 epochs.

In contrast, the $SO(2)$ GeoVAE avoids this issue through its effective coupling. By forcing the network to process both 3D geometry and sequence semantics simultaneously throughout the training, the KAE of the $SO(2)$ model decreases smoothly, leading to a stable and efficient structural convergence.

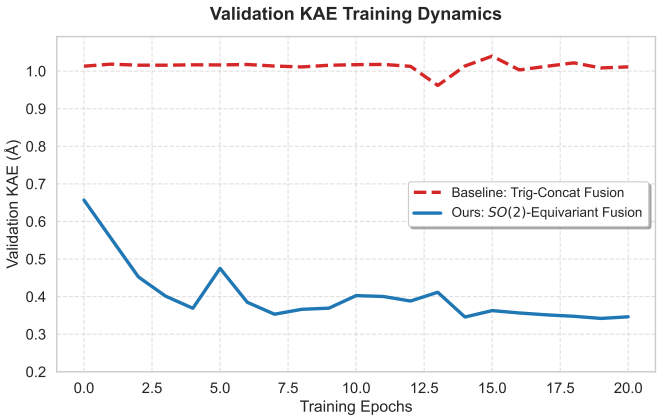

Figure 8: **Structural optimization efficiency.** Comparison of the validation KAE descent between the two fusion strategies. The Trig-Concat baseline gets trapped in an optimization plateau near 1.0 Å because it ignores spatial features, whereas the $SO(2)$ model demonstrates a smooth and stable structural convergence.

E.4 Quantitative Comparison of Sequence–Torsion Fusion Strategies

To further validate the effectiveness of the $SO(2)$-aware fusion mechanism, we extend the comparison beyond Trig-Concat by introducing a stronger periodic fusion baseline, **Fourier–FiLM** [39]. Specifically, each torsion angle χ_i is encoded using $K = 4$ Fourier frequencies, $[\sin(k\chi_i), \cos(k\chi_i)]_{k=1}^{K}$, and mapped to feature-wise scale and shift vectors $\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i \in \mathbb{R}^{256}$. These vectors modulate the sequence features through

$$\widetilde{\mathbf{h}}_i = (1 + \boldsymbol{\gamma}_i) \odot \mathbf{h}_i + \boldsymbol{\beta}_i. \tag{17}$$

We replace the $SO(2)$-aware fusion module in both the encoder and decoder while keeping the remaining GeoVAE architecture, geometric losses, and training settings unchanged.

Table 9: Comparison of sequence–torsion fusion strategies in GeoVAE on the GOLD Benchmark.

Fusion Strategy	**Angle MSE (↓)**	**Coord MSE (↓)**	**KAE (Å, ↓)**
Trig-Concat	0.1416	0.3474	1.0110
Fourier–FiLM	0.0173	0.0517	0.3659
$SO(2)$**-aware**	**0.0109**	**0.0287**	**0.2466**

As shown in Table 9, Fourier–FiLM substantially improves geometric reconstruction over Trig-Concat, indicating the importance of periodic angle encoding and adaptive feature modulation. Nevertheless, the $SO(2)$-aware fusion consistently achieves the lowest angle MSE, coordinate MSE, and KAE. These results suggest that explicitly coupling sequence features and torsion angles through an $SO(2)$-structured representation provides a more effective inductive bias than conventional periodic feature

fusion. Together with the perturbation and training-dynamics analyses above, this comparison further supports the effectiveness of the proposed $SO(2)$-aware architecture for joint sequence–geometry modeling.

F Discussion on Latent Dimension

In image-based Latent Diffusion Models (LDMs), there is a widely discussed trade-off: higher latent dimensionality (d) yields better VAE reconstruction fidelity but often increases the difficulty of training the downstream diffusion model [40]. However, in our RNA inverse folding task, we observed empirical phenomena that somewhat differ from the findings of conventional image-domain findings. This section discusses the interplay between GeoVAE information compression, DiT generative bandwidth, and training dynamics.

F.1 Decoupling Reconstruction Fidelity from Generative Capacity

Table 10: **Evaluation of latent dimension (d) scaling.** We compare the GeoVAE's upper-bound reconstruction capacity with the actual generative performance of Latent Diffusion Model. To isolate the impact of latent dimensionality, all generative metrics are evaluated on the **GOLD Benchmark under the unguided setting**. GeoVAE achieves near-perfect local reconstruction even at $d = 4$, indicating a low intrinsic physical dimension. However, Latent Diffusion Model generation peaks at $d = 8$. Further scaling up to $d = 64$ plateaus without causing severe performance degradation. Best generative results are highlighted in **bold**.

	GeoVAE (Reconstruction)		LDM (Generation)		
Latent Dim (d)	**Seq Recovery (↑)**	**KAE (↓)**	**Seq Recovery (↑)**	**SS Validity (↑)**	**Clash Score (↓)**
$d = 4$	0.9998	0.2465	0.5957	0.8747	5.41
$d = 8$	0.9996	0.2466	**0.6249**	**0.9040**	**4.51**
$d = 16$	0.9997	0.2397	0.6240	0.8955	4.54
$d = 32$	**0.9999**	0.2406	0.6249	0.9033	4.58
$d = 64$	0.9998	**0.2396**	0.6207	0.8924	4.63

As shown in Table 10, compressing the latent space to $d = 4$ maintains near-perfect GeoVAE reconstruction accuracy (Sequence Recovery ≈ 1.0, KAE ≈ 0.24Å). We attribute this to the extremely low *intrinsic manifold dimension* of RNA local interactions. The physical degrees of freedom of a single nucleotide consist solely of a discrete identity state (˜2 bits) and backbone torsion constrained by steric hindrance. A continuous 4-dimensional complex space provides a state capacity that exceeds this local information bottleneck.

However, although $d = 4$ satisfies the capacity required for VAE, the generative performance of the LDM still improves at $d = 8$. This phenomenon highlights the decoupling of local reconstruction capacity and the complexity of diffusion. In the VAE phase, forcing features into an aggressively compressed $d = 4$ space results in a highly entangled and densely packed latent manifold. Although the VAE decoder can deterministically map these entangled representations back to structures, the diffusion model faces a harder optimization task.

During the iterative denoising process, the diffusion model must approximate the score vector field ($\nabla_z \log p_t(z_t)$) to construct a generative trajectory [46]. In an over-compressed $d = 4$ space, the latent manifold is steeply non-linear, making the score function highly sensitive to Gaussian noise perturbations [51]. Expanding the dimensionality to $d = 8$ effectively unrolls the latent manifold into a slightly higher-dimensional space [4]. This unrolling reduces feature entanglement and provides a smoother score field for the diffusion model to approximate, ultimately stabilizing the generative spatial routing without incurring the curse of dimensionality.

F.2 Robustness to Latent Dimension Scaling

To explore the limits of latent dimensional scaling, we evaluated hyper-dimensional latent up to $d = 64$. In visual domains, over-parameterized latent typically encode unstructured high-frequency artifacts, leading to KL divergence explosion and catastrophic generative degradation [40]. However,

in RNA inverse folding task, our model exhibits a robust performance plateau without significant degradation in generative performance (Table 10).

In standard continuous generation tasks (e.g., image synthesis), scaling up the latent dimension typically exacerbates the rate-distortion tradeoff [40] in VAEs. To achieve better reconstruction results by utilizing higher dimensions, the model must encode more information, which significantly increases the KL divergence. This creates a severe mismatch with the standard Gaussian prior, which ultimately affects the quaility of downstream generative sampling. Conversely, if the KL penalty is strictly enforced, the model often suffers from severe posterior collapse, rendering the extra dimensions useless.

However, the training curves (Figure 9(a)) reveal a fundamentally different behavior in GOLD. Although channel capacity increased, the KL divergence loss ($\mathcal{L}_{KL}$) of the $d = 64$ model does not inflate; rather, it converges stably to a magnitude comparable to the baseline $d = 8$ setting while maintaining high-fidelity reconstruction.

We attribute this harmony to the inherently low intrinsic dimensionality of RNA structures. Unlike image pixels, RNA conformations are strictly governed by rigid physical and topological rules. Because our $SO(2)$-equivariant manifold efficiently encapsulates these geometric dependencies within a low-dimensional subspace, the excess latent dimensions in the $d = 64$ model are not forced to encode complex variances. Mathematically, this implies that the inactive dimensions contribute virtually zero to the cumulative KL penalty. Consequently, GOLD can safely scale its channel capacity to facilitate a smoother diffusion process without compromising the critical prior alignment required for generation.

To prove this assumption, we conducted a comparative statistical analysis in the empirical variance of the encoded latent means ($\mu(x)$) on the GOLD benchmark training set (Figure 9(b)). In variational inference, if the encoder captures meaningful mutual information, the predicted means $\mu(x)$ will vary on different input samples, yielding an empirical variance greater than zero. Conversely, posterior collapse occurs when the encoder outputs a constant mean ($\mu(x) \to 0$) independent of the input, causing the empirical variance to approach zero.

By independently sorting the variances of both the $d = 8$ and $d = 64$ models, a striking geometric pattern emerges: the variance profile of the active dimensions in the $d = 64$ model almost perfectly mirrors the intrinsic profile of the $d = 8$ model. The additional dimensions encode zero mutual information, with the empirical variance of their means collapsing to 0 (red dashed line).

This finding is consistent with the phenomenon of **Partial Posterior Collapse** [6, 34] in variational inference. Governed by the ELBO objective trade-off, the KL penalty actively suppresses the use of redundant capacity unless strictly necessary for reconstruction. Because RNA forms a bounded, low-frequency physical system with limited intrinsic degrees of freedom, the GeoVAE only requires a small number of core dimensions to encapsulate the intrinsic 3D geometric manifold. The surplus channels are disabled, causing the encoder to output constant zero means for those channels. Consequently, the downstream LDM merely treats these collapsed dimensions as meaningless noise, and its linear projection layers seamlessly attenuate them. This analysis confirms the robustness of our framework to latent dimension.

G Extended Ablation Studies and Generative Evaluation

In this section, we provide additional analyses of GOLD, including the advantages of the unified latent manifold over raw-space co-diffusion, the effectiveness of gradient guidance compared to direct secondary-structure conditioning, and the one-to-many generative properties of RNA design. We further investigate the interaction and sensitivity of the two physical guidance objectives to better understand their effects on the sampling process.

G.1 Unified Latent Manifold vs Raw-Space Co-Diffusion

Recently, several pioneering studies have explored the co-design of RNA sequences and structures by directly applying diffusion models to the raw data spaces [35, 32]. These methods typically combine discrete categorical diffusion for sequences and continuous Gaussian diffusion for structures, linking them primarily through multi-task loss functions. Although these models are promising, we cannot

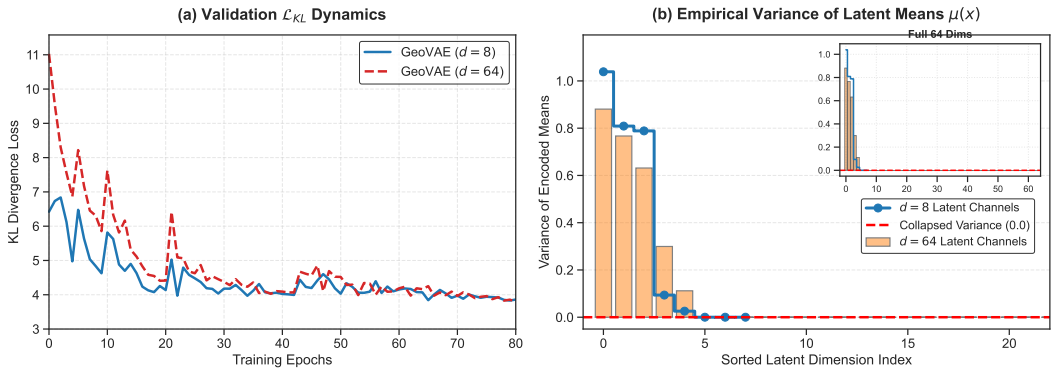

Figure 9: **Comparative evidence of Partial Posterior Collapse.** (a) The KL divergence loss remains stable across 80 epochs even when channel capacity is scaled $d = 8$ to 64. (b) Comparative analysis of the sorted empirical variance of latent means ($\mu(x)$). The **main plot** provides a zoomed-in view of the active dimensions, revealing that the variance profile of the over-parameterized $d = 64$ model perfectly mirrors the intrinsic active dimensions of the baseline $d = 8$ model. This confirms that all remaining redundant channels collapse to zero (red dashed line). A zero variance of encoded means indicates that the encoder outputs a constant regardless of the input, proving that these excess dimensions encode no mutual information and are entirely suppressed by the posterior collapse mechanism.

deploy them directly as baseline models for two main reasons. First, our task focuses on inverse folding, which requires generating sequences conditioned on a given 3D backbone, whereas these co-design methods target different generation scenarios. Second, their source codes are currently not publicly available.

Raw-Space Co-Diffusion. To analyze the structural impact of the latent manifold, we implemented a custom **Raw-Space Co-Diffusion Baseline**, as illustrated in Figure 10. To ensure a strictly controlled and fair comparison, this baseline retains the exact same generative backbone (GVP + DiT) as GOLD, but completely omits the GeoVAE. Instead of operating within a compressed manifold, it diffuses directly in the data space through the following mechanisms:

- **Discrete Sequence Branch**: Applies categorical diffusion directly to the 4-letter nucleotide tokens, utilizing the exact same discrete transition kernels as our latent formulation.
- **Continuous Angle Branch**: Performs standard Gaussian diffusion on the trigonometric representations of the side-chain angles (χ) within the raw coordinate space.
- **Input-side Fusion**: At each denoising timestep, the noisy sequence embeddings and noisy continuous angle vectors are simply concatenated at the input-side before being processed by the joint GVP-Transformer layers.
- **Joint Optimization**: The network is trained end-to-end to simultaneously minimize the raw-space data objectives, specifically cross entropy for sequence classification and MSE for angle regression.

Experimental Results. We evaluated both models on the internal GOLD Test Benchmark. To explicitly assess the native generation quality without relying on external folding oracles, we report the Sequence Recovery alongside two intrinsic structural metrics: SS Validity and Clash Score. As shown in Table 11, while the Raw-Space Baseline achieves reasonable performance, its performance consistently worse than **GOLD (Base)** across all metrics. This performance gap indicates that simply combining parallel losses is insufficient for capturing the complex interdependencies of RNA. The unified latent manifold provides a smoother, physically entangled representation that effectively bridges the modality gap, natively suppressing steric clashes and improving topological pairing.

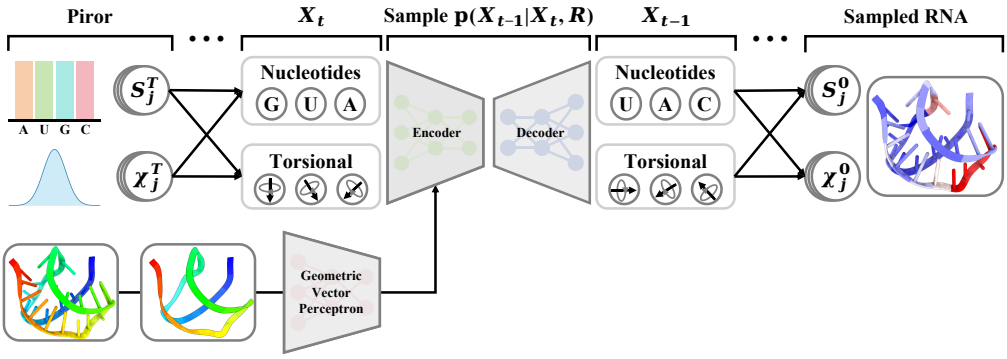

Figure 10: **Architecture of the Raw-Space Co-Diffusion.** Unlike GOLD's unified latent manifold, this experiment performs independent diffusion processes (categorical and Gaussian) in their respective raw data spaces, fusing the noisy modalities solely through input-level concatenation.

Table 11: Ablation study comparing the Unified Latent Manifold (GOLD) against the Raw-Space Co-Diffusion baseline on the GOLD Test Benchmark. All models are evaluated in the unguided (Base) setting using native outputs.

Architecture	Seq Recovery (↑)	SS Validity (↑)	Clash Score (↓)
Raw-Space Baseline	0.6036	0.8657	6.53
GOLD (Base)	**0.6249**	**0.9040**	**4.51**

G.2 Input Feature vs Gradient Guidance

In our Dual-Guided framework, the generative trajectory is guided by the target 2D secondary structure (SS). To analyze the effectiveness of our gradient-based sampling mechanism from the availability of SS data, we conducted an ablation study to explore different methods of incorporating topological information.

Ablation Setup: SS as an Input Feature. We constructed a variant, **GOLD (Base w/ Input SS)**, which integrates the target SS graph directly into the initial spatial conditioning phase. Specifically, we explicitly added the target base-pairing edges to the spatial graph. In the edge attributes, edges connecting pairs of nucleotides were assigned a distinct binary indicator and this enriched graph was then processed by the GVP. In this setup, the model "sees" the exact same target topology as the guided version but relies entirely on implicit neural message passing to satisfy these constraints.

Experimental Results and Analysis. As shown in Table 12, simply providing the target SS as input feature results in only a slight improvements compared to the GOLD (Base). This indicates that the network struggles to autonomously translate discrete input edge features into strict, long-range sequence pairing semantics during the latent denoising process. In contrast, **GOLD (SS-Guided)**, which enforces the exact same SS information through active gradient penalization ($\omega_{ss} = 1.0$) at each sampling step, achieves a larger leap in both Sequence Recovery and SS Validity. This proves that the superior performance of our method stems from the explicit, dynamic mechanism of gradient guidance, rather than a trivial leakage of topological information.

G.3 One-to-Many Generative Evaluation

RNA inverse folding is inherently a one-to-many generation task, as multiple distinct sequences may be compatible with the same target structure. Therefore, conventional recovery metrics alone cannot fully characterize the generative capability of a design method. To provide a more comprehensive evaluation, we additionally assess sequence novelty, structural success rate as a proxy for designability, and intra-target sequence diversity, following evaluation practices in related studies [24, 28, 42].

Table 12: Ablation study on the utilization of Secondary Structure (SS) information evaluated on the GOLD Test Benchmark. Applying the target SS merely as an input edge feature yields minimal gains, demonstrating that the dynamic gradient intervention (SS-Guided) is the fundamental driver of topological accuracy.

Model Variant	**Seq Recovery (↑)**	**SS Validity (↑)**	**Clash Score (↓)**
GOLD (Base)	0.6249	0.9040	4.51
GOLD (Base w/ Input SS)	0.6253	0.9045	4.48
GOLD (SS-Guided)	**0.6267**	**0.9343**	**4.32**

Evaluation Metrics. Sequence novelty measures the dissimilarity between generated sequences and the training set:

$$\text{Novelty}_{seq} = 1 - \frac{1}{n}\sum_{i=1}^{n}\max_{t\in\mathcal{T}}\text{Id}(s_i, t), \tag{18}$$

where $\mathcal{T}$ denotes the training set, s_i is a generated sequence, and $\text{Id}(s,t)$ is the sequence identity computed using Needleman–Wunsch global alignment.

Structural success rate measures the proportion of generated sequences whose predicted structures satisfy a given scRMSD threshold:

$$\text{SuccessRate}(\tau) = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}[\text{scRMSD}_i \le \tau], \tag{19}$$

where $\tau \in \{2, 5, 8, 10\}$ Å. While 2 Å provides a strict structural criterion, we also consider broader thresholds to account for RNA flexibility and the limitations of current RNA structure prediction tools.

To evaluate sequence diversity, we compute the internal diversity (IntDiv) among the eight sequences generated for each target:

$$\text{IntDiv}(G_X) = 1 - \frac{1}{m_X^2}\sum_{i,j}\text{Sim}(s_i, s_j), \tag{20}$$

where G_X is the set of $m_X = 8$ generated sequences for target X, and $\text{Sim}(s_i, s_j) = 1 - d_H(s_i, s_j)/L$ denotes normalized Hamming similarity. We report the average IntDiv across all test targets.

Table 13: Sequence novelty, structural success rate, and intra-target diversity on the GOLD Benchmark.

Method	**Novelty**	**Success Rate (↑)**				**Diversity**
		2Å	**5Å**	**8Å**	**10Å**	
PiFold	0.2858	0.0327	0.1763	0.3184	0.3752	0.1148
RDesign	**0.3291**	0.0468	0.2227	0.3853	0.4566	0.0788
gRNAde	0.3254	0.0356	0.1938	0.3519	0.4298	0.0842
RiboDiffusion	0.2989	0.0668	0.3742	0.5189	0.6102	**0.1264**
SOLD	0.2949	0.0585	0.3913	0.5569	0.6321	0.1121
GOLD (Dual-Guidance)	0.3183	**0.0724**	**0.4242**	**0.5935**	**0.6670**	0.1015

Results and Analysis. As shown in Table 13, GOLD achieves the highest structural success rate across all four thresholds, indicating improved structural designability. Meanwhile, GOLD maintains competitive sequence novelty and substantial intra-target diversity, suggesting that its improved structural success does not simply arise from reproducing training sequences. Although its diversity is moderately lower than that of some competing methods, this may reflect a trade-off between structural consistency and sequence diversity under stronger physical guidance. Overall, these results further demonstrate the effectiveness of GOLD for one-to-many RNA inverse folding beyond conventional recovery metrics.

Table 14: Interaction between secondary-structure and clash guidance on the GOLD Benchmark. Each cell reports Sequence Recovery / SS Validity / Clash Score (↑ / ↑ / ↓). The default configuration is highlighted in bold.

$\omega_{\text{ss}} \backslash \omega_{\text{clash}}$	0	0.05	0.10	0.50	1.00
0	0.625/0.904/4.51	0.636/0.901/3.11	0.638/0.903/2.76	0.635/0.906/2.28	0.626/0.894/2.77
1	0.627/0.934/4.32	0.637/0.932/3.06	**0.638/0.929/2.67**	0.633/0.928/2.31	0.627/0.927/2.74
2	0.626/0.936/4.43	0.634/0.934/3.08	0.636/0.931/2.69	0.632/0.930/2.27	0.621/0.931/2.81
3	0.627/0.938/4.38	0.635/0.936/3.10	0.637/0.932/2.72	0.630/0.933/2.32	0.624/0.932/2.78

G.4 Interaction and Sensitivity of Physical Guidance

To better understand the behavior of dual-guided sampling, we investigate the interaction between secondary-structure and clash guidance, as well as the sensitivity of the reverse diffusion process to their respective guidance strengths.

Interaction Between Guidance Objectives. We conduct a two-dimensional parameter sweep over the secondary-structure guidance weight $\omega_{\text{ss}} \in \{0, 1, 2, 3\}$ and clash guidance weight $\omega_{\text{clash}} \in \{0, 0.05, 0.10, 0.50, 1.00\}$. Each configuration is evaluated using Sequence Recovery, SS Validity, and Clash Score to examine whether the two guidance objectives cooperate or compete.

As shown in Table 14, the two guidance objectives are largely complementary under moderate weights. Secondary-structure guidance primarily improves topological validity, whereas clash guidance effectively reduces steric conflicts and can also improve sequence recovery. Their joint application largely preserves both benefits without substantially compromising sequence recovery. However, excessively strong clash guidance leads to diminishing structural improvements and eventually degrades sequence recovery, suggesting the importance of balancing the two constraints.

Guidance Sensitivity and Reverse-Process Deviation. To further understand the effects of guidance strength, we quantify how much each guidance objective perturbs the unguided reverse diffusion process. At reverse step $t \to s$, we consider the unguided and guided transition distributions:

$$p_0(z_s \mid z_t) = \mathcal{N}(\mu_t^0, \sigma_t^2 I), \qquad p_g(z_s \mid z_t) = \mathcal{N}(\mu_t^g, \sigma_t^2 I). \tag{21}$$

Since the two transitions share the same covariance, their step-wise conditional KL divergence is

$$D_t = D_{\text{KL}}\left[p_g(z_s \mid z_t) \| p_0(z_s \mid z_t)\right] = \frac{\|\mu_t^g - \mu_t^0\|_2^2}{2\sigma_t^2}. \tag{22}$$

Both transitions are evaluated at the same current guided state z_t, allowing us to isolate the local perturbation introduced by guidance. We accumulate this divergence along the guided trajectory and normalize it by RNA length L and latent dimension d:

$$KL_{\text{path}} = \sum_t \frac{D_t}{Ld}. \tag{23}$$

The average KL_{path} over evaluation trajectories provides an operational measure of deviation from the learned reverse process, rather than an exact geometric distance to the latent manifold. We vary one guidance coefficient at a time while setting the other to zero.

As shown in Table 15, secondary-structure guidance introduces relatively small deviations from the unguided process. Moderate guidance captures most of the improvement in SS validity, while further increasing the weight yields diminishing returns with little effect on sequence recovery. In contrast, clash guidance produces substantially larger trajectory deviations. Moderate clash guidance reduces steric conflicts and improves sequence recovery, whereas excessive guidance sharply increases KL_{path}, accompanied by saturated or deteriorating structural improvements and reduced sequence recovery. These results suggest that overly strong guidance can perturb the learned reverse dynamics beyond a beneficial correction regime. Accordingly, we select the default guidance weights by balancing structural validity, steric feasibility, and sequence recovery.

H Analysis of Dual-Guidance Dynamics

In the main text, global statistical evaluations demonstrated that combining topological and spatial guidance improves the quality of generation. To elucidate the specific physical dynamics underlying

Table 15: Guidance sensitivity and reverse-process deviation under different guidance strengths.

Guidance	Weight	$\mathrm{KL}_{\mathrm{path}}$	SS Validity ↑	Clash Score↓	Seq Recovery ↑
SS	0	0	0.904	–	0.625
SS	1	0.075	0.934	–	0.627
SS	2	0.541	0.936	–	0.626
SS	3	0.905	0.938	–	0.627
Clash	0	0	–	4.51	0.625
Clash	0.05	10	–	3.11	0.636
Clash	0.10	28	–	2.76	0.638
Clash	0.50	515	–	2.28	0.635
Clash	1.00	3046	–	2.77	0.627

the sampling process, this section provides an in-depth case study. We analyze how the guidance parameters ω_{ss} and ω_{clash} influence its local spatial frame.

H.1 Topological Guidance (ω_{ss})

Although the main text establishes that ω_{ss} improves global secondary structure validity, it is important to examine how this guidance alters the generative trajectory at the micro-scale. As illustrated in Figure 11, in the unguided state ($\omega_{ss} = 0$), relying solely on the learned prior of the latent manifold is sometimes insufficient to fully close complex structural motifs, such as the stem of target `2K5Z_A`. The absence of explicit spatial pairing constraints leaves the stem partially open, resulting in a pairing validity of 81.82%.

By introducing the topological guidance weight ($\omega_{ss} = 1.0$), the model successfully forms the stem. More importantly, detailed sequence analysis reveals that this structural correction is achieved through context-aware sequence adjustment rather than simply memorizing the native sequence. For example, at the critical pairing site (residues 5 and 24), the unguided model samples an unpaired C-A combination, which physically prevents the stem from forming a stable conformation. Under the gradient guide of $\nabla\mathcal{E}_{ss}$, the sampling trajectory dynamically shifts these specific positions into a stable Watson-Crick pair (C-G). This micro-level correction demonstrates that the guidance gradient effectively reshapes the local sequence probability distribution, actively guiding the generation into valid physical conformations that conforms the target topology.

H.2 Clash Guidance (ω_{clash})

Visualization Protocol. To illustrate the microscopic dynamics of latent spatial guidance, the 3D conformations presented in Figure 12 and main text Figure 3 are generated using the same protocol:

- **Raw Generative Output:** The 3D coordinates are directly produced by our diffusion model's sampling trajectory. We intentionally bypassed any external energy minimization or structural relaxation algorithms, ensuring the visualizations reflect the model's unrelaxed generative capabilities under specific guidance weights.
- **Topological Backbone Grafting:** Auxiliary backbone atoms absent from our coarse-grained representation were grafted from native structures solely to facilitate continuous PyMOL ribbon rendering, leaving all predicted $C4'$ and side-chain coordinates completely unaltered.
- **Standardized Highlighting:** All states are rendered using a fixed camera matrix. Atomic steric overlap, defined as instances where the distance between two non-bonded atoms falls below a threshold of 2.0 Å, are highlighted using **red transparent van der Waals spheres** to signify spatial penetration. When clashes are resolved through an appropriate guidance weight, these red spheres disappear, revealing cleanly packed side-chain conformations.

Effects of Excessive Penalization. As illustrated in Figure 12, scaling the clash guidance reveals a non-linear optimization trade-off. A moderate guidance weight ($\omega_{clash} = 0.1$) effectively resolves atomic penetrations while maintaining high sequence recovery. However, applying an excessively high spatial penalty ($\omega_{clash} = 1.0$) leads to a decline in generative quality and manifests as two different behaviors:

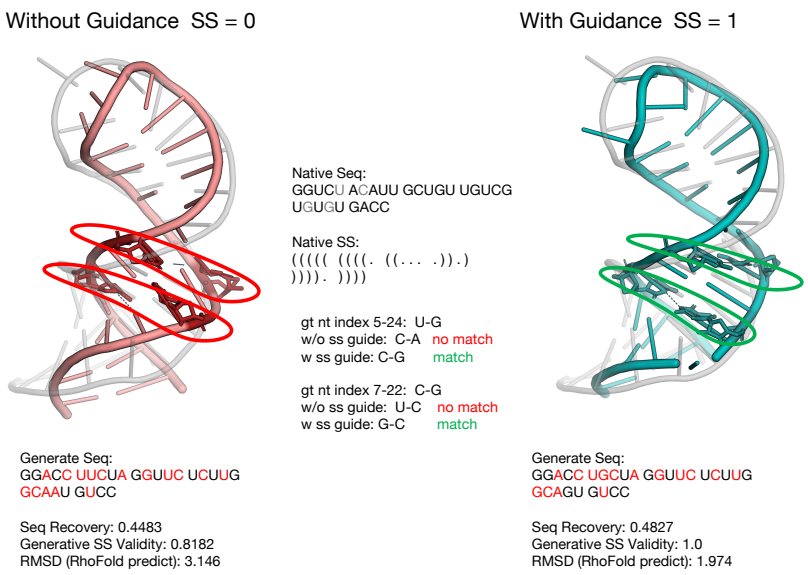

Figure 11: Observation of topology and sequence intervention under Topological Guidance (ω_{ss}) for target `2K5Z_A`. **Left ($\omega_{ss} = 0$):** Without guidance, the generative trajectory leaves the stem partially open, producing unviable base pairs (highlighted in red). **Right ($\omega_{ss} = 1.0$):** The guidance gradient effectively corrects the pairing network by shifting the local sequence probability toward valid Watson-Crick pairs (highlighted in cyan). This intervention successfully closes the stem with 100% validity, illustrating how the model resolves physical conflicts without simply relying on sequence memorization.

- **Behavior I: Sequence-Structure Trade-off (e.g., Target `5WNV_B`).** Under a heavy spatial penalty, the model tends to sacrifice sequence identity to satisfy steric constraints. This altered the sequence prediction, causing the recovery rate to drop from 0.73 to 0.53, and adopts extended side-chain conformations to create spatial gaps. Although this optimization reduces the clash score to zero, the final topology noticeably deviates from the native structure, resulting in an RMSD of 12.47 Å.
- **Behavior II: Gradient-Induced Manifold Deviation (e.g., Target `6C63_A`).** In densely packed regions, an excessively repulsive weight produces unusually large gradients during latent sampling. These significant updates can push the latent trajectory z_t outside the learned geometric prior distribution and decoding these out-of-distribution states may disrupt the spatial projection mechanism. Consequently, the collision score increases significantly to 24.32, while the sequence recovery drops to 0.44.

These observations suggest that our dual-guided mechanism serves as a physical regularizer. The selected optimal weight ($\omega_{clash} = 0.1$) maintains a delicate balance between adhering to the prior geometric manifold and satisfying explicit spatial constraints.

Extensibility of Physical Guidance. Beyond secondary-structure pairing and steric clashes, GOLD provides a flexible framework for incorporating additional RNA-specific physical interactions. By differentiably reconstructing atomic-level nucleobase coordinates from soft sequence probabilities and glycosidic torsion angles, GOLD allows new physical constraints to be formulated as differentiable energy functions and integrated into the reverse diffusion process without retraining the generative model.

For example, **base-stacking interactions**, which are important for RNA structural stability, could be modeled using inter-base distances, relative plane orientations, lateral offsets, and geometric overlap. Unlike clash guidance, which primarily penalizes unfavorable atomic overlaps, stacking potentials could encourage favorable spatial arrangements between nucleobases. Similarly, non-canonical base pairing, hydrogen-bond geometry, and base–phosphate interactions could be incorporated through

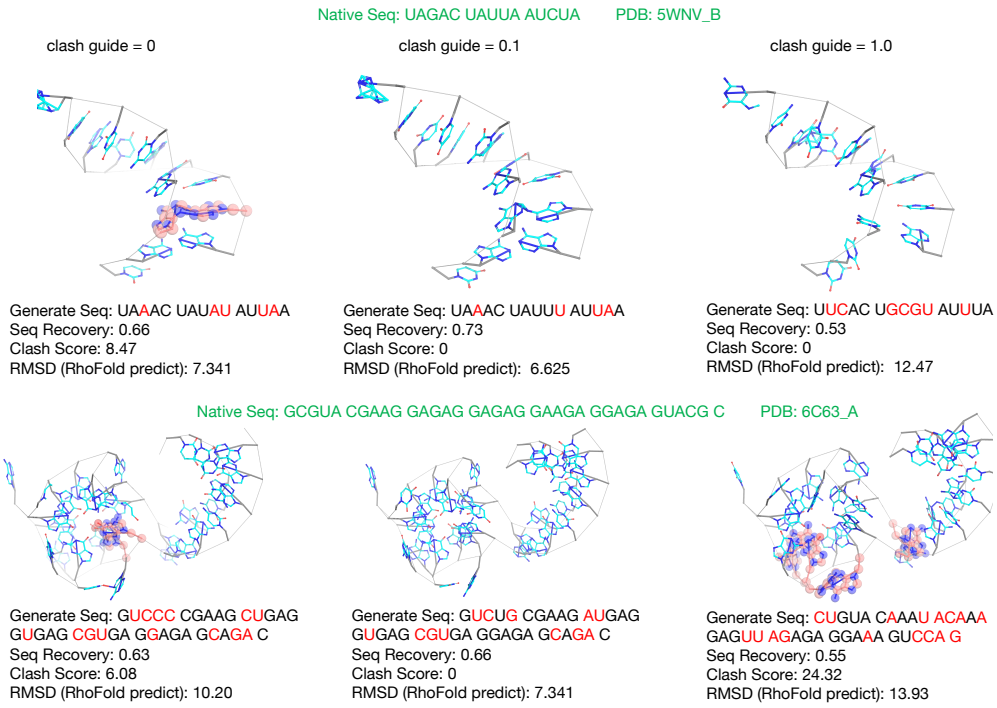

Figure 12: Ablation of spatial guidance intensity (ω_{clash}) on targets `5WNV_B` and `6C63_A`. The data demonstrates that while an appropriate weight (0.1) resolves physical clashes and preserves sequence recovery, excessive penalization (1.0) leads to generative degradation.

additional energy terms. Although these extensions have not yet been systematically evaluated, they demonstrate the potential of GOLD to accommodate richer physical constraints for RNA inverse folding.

I Extended Case Studies

To supplement the evaluations in the main text, this section provides detailed structural comparisons. We present two PDB targets, evaluating GOLD against five baselines: PiFold, RDesign, gRNAde, RiboDiffusion, and SOLD.

I.1 Case Study I

For the complex target `4KNQ_B` (Figure 13), all five baselines fail to orient local fragments globally (RMSD > 16.0 Å), it highlights the inherent vulnerability of models that remain agnostic to side-chain geometry, which is essential for translating local accuracy into global structural coherence. In contrast, **GOLD** achieves near-native structural integrity (RMSD **2.929 Å**), demonstrating that our joint diffusion process effectively captures the long-range spatial dependencies required for complex 3D assembly.

I.2 Case Study II

Target `6Q8V_A` is used to evaluate the accuracy of sequence-to-structure prediction. Although baseline methods such as RDesign achieve good sequence recovery, their 3D structures are often mismatched. **GOLD** overcomes this issue, achieving **0.94** sequence recovery, **1.0** SS validity, and a highly accurate 3D conformation.

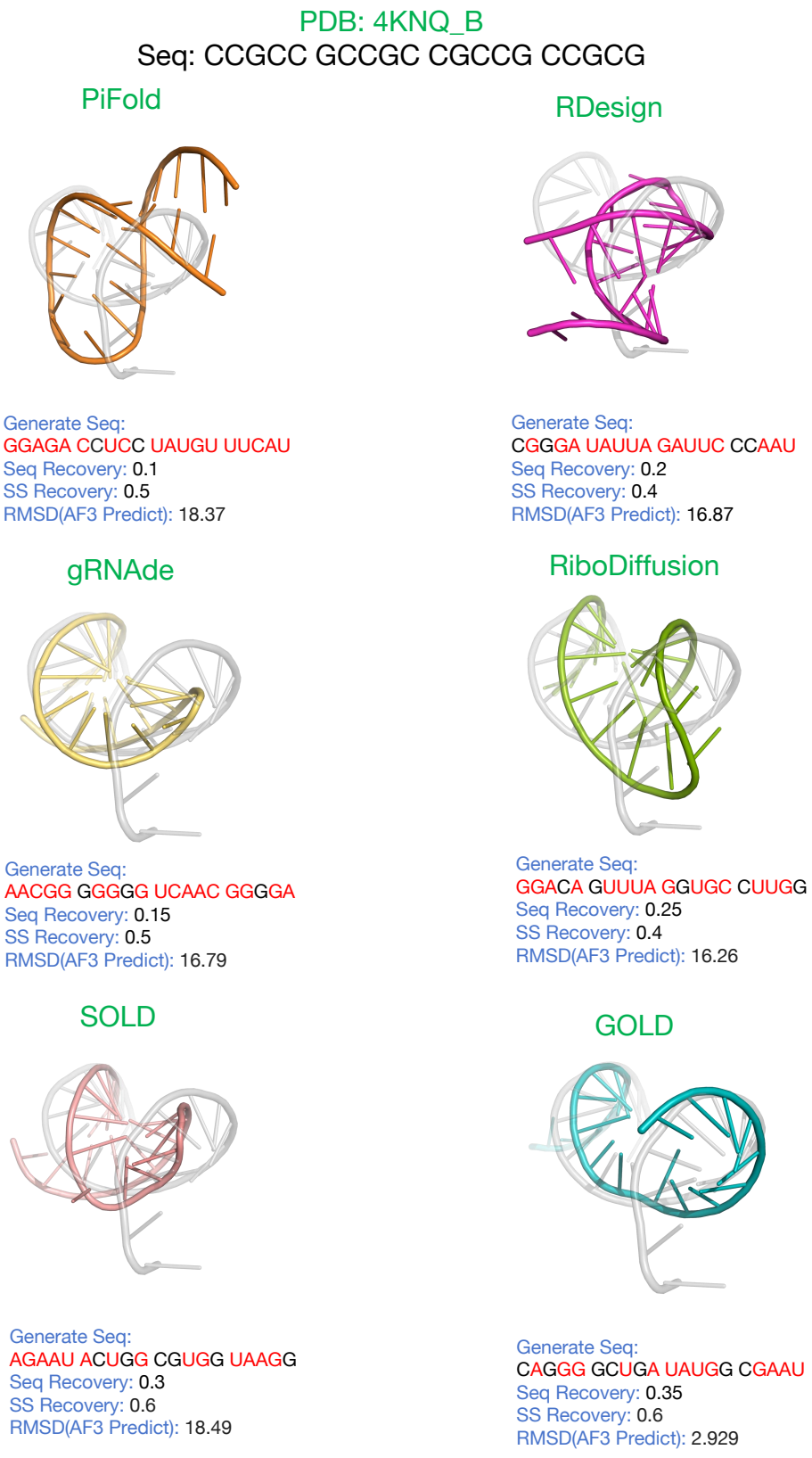

Figure 13: **Comprehensive Comparison on Target** `4KNQ_B`**.** Baselines exhibit severe topological collapse, whereas GOLD successfully recovers the global fold.

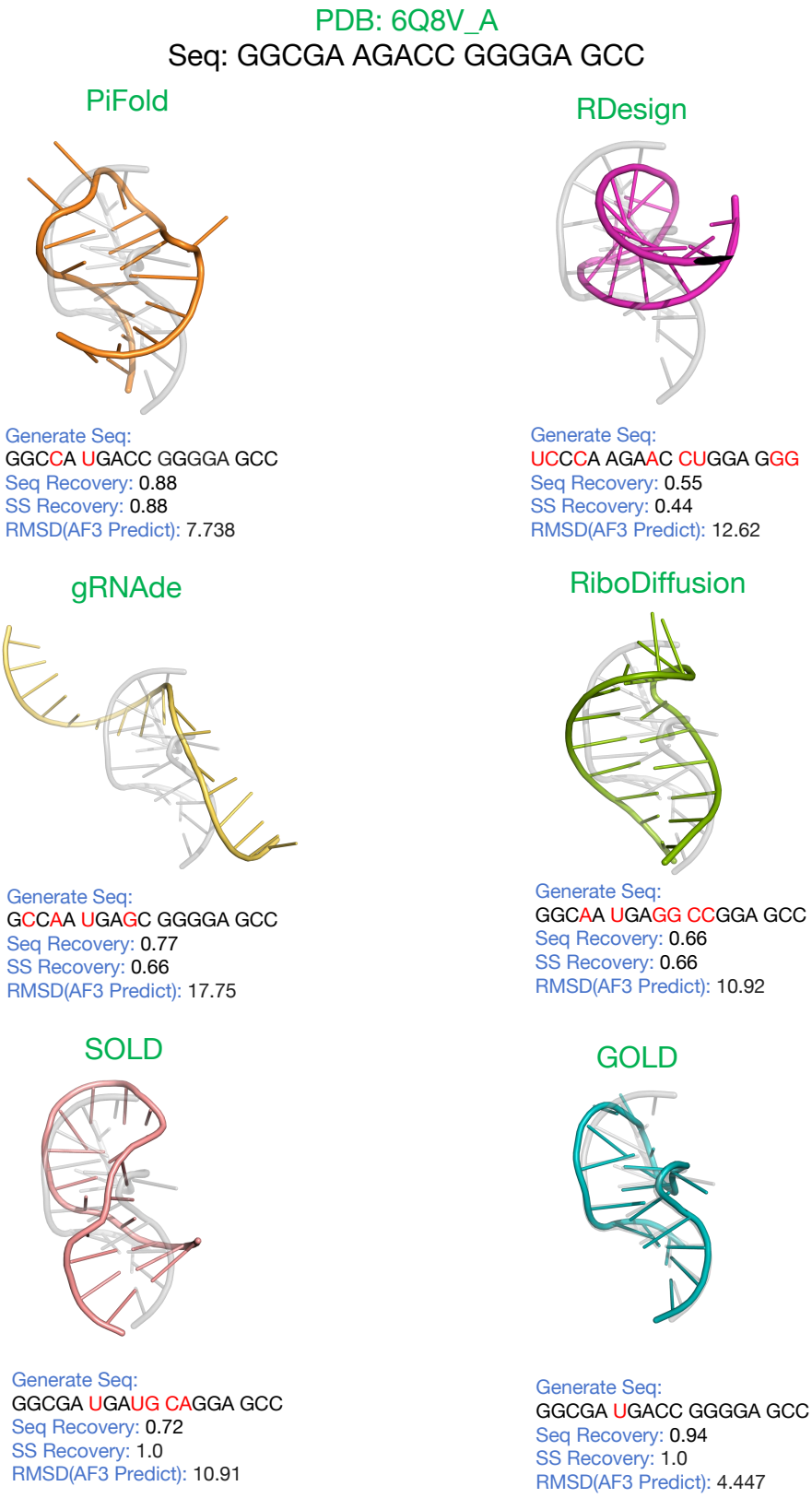

Figure 14: **Comprehensive Comparison on Target** `6Q8V_A`**.** GOLD demonstrates superior sequence-structure fidelity, achieving 100% SS validity and the lowest RMSD (4.4Å) compared to baselines.

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: In the abstract and introduction, we clearly state the contributions of our paper.

   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: In main text Conclusion, we clearly describe the limitations of the work.

   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: [N/A]

Justification: This paper does not include theoretical results.

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: Appendix C provides the algorithm's details of hyperparameters and workflows. The complete code and data are made public as `https://github.com/darkflash03/GOLD`.

   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: We clearly discussed the training datasets and other details in Appendix C.

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: The paper specify all the training and test details in Appendix B.

   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: For both our models and the baselines, we generate multiple outputs for each test case and report the average performance across all samples as shown in main text experiment (main results).

   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: The paper provides sufficient information on the computer resources in Appendix B.3 and Appendix D.2.

   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: The research in this paper conforms in every respect with the NeurIPS Code of Ethics.

   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: The paper discussed potential societal impacts in main text conclusion.

    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: The paper poses no such risks.

    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: The existing assets are properly cited and credited.

    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: The paper release the GOLD Benchmark and code at `https://github.com/darkflash03/GOLD`.

    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: The paper does not involve crowdsourcing nor research with human subjects.

    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 paper does not involve crowdsourcing nor research with human subjects.

    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 were not used as an important, original, or non-standard component of the core methods in this research. They were solely utilized for writing and editing assistance.

    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.