Antibody Generation via Redistributed Latent Diffusion

Andrey Shevtsov [*1] **Viacheslav Meshchaninov** [*2 3] **Pavel Strashnov** [*1] **Dmitry Vetrov** [2]

Abstract

Generating antibody sequences is challenging because they combine conserved framework regions with hypervariable loops. Latent diffusion is attractive for this task since it enables flexible conditioning and bidirectional generation. But standard approaches fail. Global noise schedules treat all positions equally, so models learn the predictable frameworks well while the diverse loops remain poorly captured. We address this by learning a latent space that redistributes information evenly, allowing standard diffusion to succeed where it previously failed. On organism-conditioned generation across six species, our approach achieves 10× lower Fréchet Distance than latent diffusion without redistribution. It supports chain-type control, loop infilling, and paired-chain generation. Validation across five protein encoders confirms the method is encoder-agnostic. These results establish latent diffusion as a practical tool for antibody sequence design.

1. Introduction

Latent Diffusion Models (LDMs) (Rombach et al., 2021) perform iterative denoising in a compressed latent space learned by a pretrained encoder. Applied to protein sequences with pretrained language model encoders, they reach strong distributional fidelity on general protein benchmarks while supporting non-autoregressive, bidirectional generation (Meshchaninov et al., 2025b).

Antibody sequences present a fundamental challenge for latent diffusion models because they contain near-zero-entropy framework regions interspersed with hypervariable CDR loops. Framework regions vary little across the antibody repertoire, whereas complementarity-determining regions (CDRs), especially CDR-H3, are highly diverse (Briney et al., 2018) because of V(D)J recombination (Tonegawa, 1983; Weitzner et al., 2015). Under a uniform global noise schedule, the model faces a trade-off: noise levels appropriate for framework regions obliterate signal in CDRs, while levels suitable for CDRs make framework regions trivially easy to denoise. Unlike autoregressive models that adjust uncertainty position by position, diffusion models must denoise the full sequence simultaneously. As a result, training loss concentrates in CDR positions, while framework positions contribute little to learning. This capacity misallocation prevents the model from learning the joint distribution over conserved and variable regions.

Recent work addresses non-uniform information content via adaptive noise schedules across pixels (Sahoo et al., 2023), bits (Dieleman et al., 2022), or time-series positions (Lee et al., 2024). These methods modify the diffusion process while keeping the representation fixed. Prior protein-sequence work has noted the challenge of high-entropy regions in antibodies (Frey et al., 2023; Atkinson et al., 2024) without diagnosing the root cause or considering representation learning as a fix. We observe that the imbalance originates in the latent space itself: encoders trained on diverse protein families preserve positional entropy variation when applied to antibodies, creating large disparities in local signal-to-noise ratio across positions.

To address this, we learn a latent representation that redistributes information more evenly across dimensions. We use COSMOS (Meshchaninov et al., 2025a), a method for learning fixed-length compressive latent spaces with explicit capacity regularization, to "smooth" the entropy landscape. By constraining the compressor to use all latent dimensions equally, COSMOS yields representations in which diffusion capacity aligns with learning difficulty across positions, removing the signal-to-noise mismatch without position-specific noise schedules.

Key contributions: *(a)* We identify capacity misallocation as a failure mode of latent diffusion on antibody sequences and ground it in per-position entropy and loss analyses (Sec. 3). *(b)* We show that learning a redistributed latent representation addresses the cause, and validate this across five encoder architectures (Secs. 4 and 5.3). *(c)* We achieve strong generation performance across six organisms, with conditional generation for CDR infilling, heavy-light chain pairing, and structure-conditioned generation (Sec. 5).

[1]AXXX, Moscow, Russia [2]Constructor University, Bremen, Germany [3]HSE University, Moscow, Russia. Correspondence to: Pavel Strashnov <strashnov@axxx.tech>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

2. Background

We formalize the incompatibility between the homogeneous assumptions of continuous-time latent diffusion and the heterogeneous nature of antibody sequences.

2.1. Latent Diffusion and Global Noise Schedules

We consider Latent Diffusion Models (LDMs) operating in a compressed latent space $\mathbf{z} = \mathcal{E}(\mathbf{x})$, where $\mathbf{z} \in \mathbb{R}^{L \times D}$ (Rombach et al., 2021). We adopt the continuous-time framework with $t \in [0, 1]$, where the forward process progressively corrupts the data $\mathbf{z}_0$ to a standard Gaussian prior $\mathbf{z}_1 \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$ (Song et al., 2020). The transition kernel is given by $q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; s(t)\mathbf{z}_0, \sigma^2(t)\mathbf{I})$, with **signal-to-noise ratio (SNR)** $\lambda(t) = s^2(t)/\sigma^2(t)$ measuring information content at timestep t (Kingma et al., 2021).Crucially, standard diffusion formulations impose a **global noise schedule**, where $\lambda(t)$ is a scalar function. This implies that the rate of signal decay is uniform across all latent dimensions. The model effectively assumes that the information content is homogeneous, or that the difficulty of denoising is independent of the position within the sequence.

2.2. Antibody Sequences and Information Density

Antibodies represent a specialized class of proteins characterized by a sharply partitioned architecture. The variable sequence $S = (s_1, \ldots, s_L)$ is divided into four structural *Framework* regions (FR1–FR4) and three *Complementarity-Determining Regions* (CDR1–CDR3) (Lefranc et al., 2003).

These regions exhibit an extreme **information density imbalance**, which we define as the spatial heterogeneity in per-position Shannon entropy. Framework regions are highly conserved, exhibiting low entropy ($H(s_i) \approx 0$ for $i \in I_{\text{FW}}$). In contrast, CDR regions—particularly CDR-H3—are hyper-variable, exhibiting high entropy ($H(s_i) \gg 0$ for $i \in I_{\text{CDR}}$) (Briney et al., 2018).

When processed by pre-trained protein language models such as ESM-2 (Lin et al., 2023), the resulting latent representation $\mathbf{z}$ preserves this entropy landscape. The encoder $\mathcal{E}$ maps the input's statistical properties into the latent space, maintaining the contrast between the low-information structural scaffolds and the high-information binding loops.

2.3. Capacity Misallocation

The incompatibility becomes concrete when we quantify the information density imbalance. Figure 1 shows the 8-gram entropy across antibody regions, computed over the human heavy chain repertoire. The contrast is stark. CDR3 exhibits 14.09 ± 0.10 bits of entropy, while framework regions range from 4.69 ± 0.01 (FR4) to 6.52 ± 0.01 bits (FR3). Since the effective number of local sequence variants scales as 2^H, this entropy gap implies that CDR3 occupies a combinatorial space $190\times$ larger than FR3 and $675\times$ larger than FR4. Adjacent positions in the same sequence

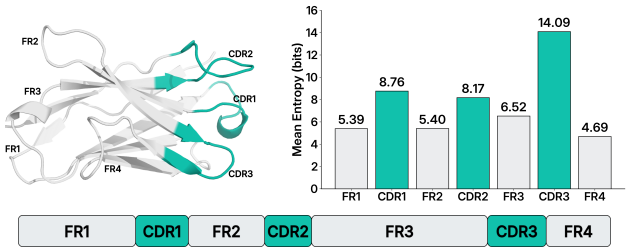


Figure 1. **Information Density Imbalance in Antibody Sequences.** *(Left)* Variable-domain structure with framework regions (FR1–FR4) and complementarity-determining regions (CDR1–CDR3). *(Bottom)* Sequence organization with segment lengths proportional to mean lengths in natural repertoires. *(Right)* Mean 8-gram Shannon entropy across human heavy chains. CDR3 (14.09 bits) exceeds FR4 (4.69 bits) by a gap corresponding to $675\times$ more combinatorial variants.

require reconstruction from distributions of fundamentally different complexity.

This heterogeneity conflicts directly with global noise schedules. At any timestep t, the schedule $\lambda(t)$ determines the signal-to-noise ratio uniformly across all positions. But uniform SNR applied to non-uniform complexity produces non-uniform learning difficulty. For framework positions, low entropy means few plausible reconstructions, so the denoising target remains identifiable even at high noise levels since prediction is trivial. For CDR positions, high entropy means the target is drawn from an exponentially larger space of plausible sequences. At the same noise level, the signal is indistinguishable from the prior, and prediction collapses to the mean.

We term this failure mode **capacity misallocation**. Because loss is averaged across positions, gradients are dominated by the high-SNR regime where frameworks provide strong, simple signal. CDR positions, trapped in a low-SNR regime throughout training, receive insufficient learning signal. This mirrors **simplicity bias** in discriminative models (Shah et al., 2020), where networks exploit easy features while ignoring complex ones.

The key insight is that this failure originates in the *representation*, not the diffusion process. Pre-trained encoders preserve the input's entropy landscape, embedding the two-order-of-magnitude complexity gap directly into the latent space. Modifying the noise schedule treats the symptom; learning a redistributed representation addresses the cause. Sec. 3 provides empirical confirmation that capacity misallocation manifests as concentrated per-position loss in CDR regions.

3. Capacity Misallocation in Latent Diffusion

The information density imbalance described in Sec. 2 predicts that latent diffusion models trained on antibody se-

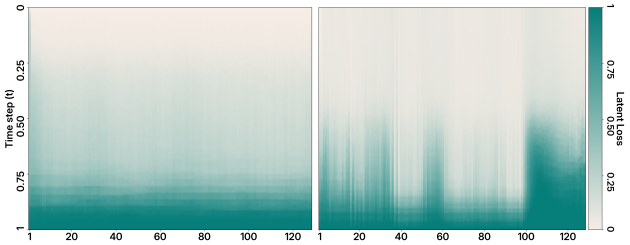


Figure 2. **Per-position diffusion loss across timesteps for ESM-2-based latent diffusion models.** *(Left)* Training on general proteins yields uniform loss across positions. *(Right)* Training on antibodies reveals capacity misallocation, with elevated loss (teal bands) concentrated in CDR regions throughout the diffusion process.

quences should struggle with high-entropy regions while easily capturing low-entropy ones. We now provide direct empirical evidence for *capacity misallocation*.

Diagnostic setup. To visualize how a diffusion model allocates its learning capacity across sequence positions, we examine the per-position loss $\mathcal{L}(i,t) = \mathbb{E}[\|\epsilon - \epsilon_\theta(z_t, t)\|_i^2]$ as a function of both position i and timestep t. This produces a heatmap where vertical bands of elevated loss indicate positions the model struggles to denoise throughout training. A well-matched model and data distribution should yield roughly uniform loss across positions at each timestep.

Evidence of misallocation. Fig. 2 compares this diagnostic for a latent diffusion model using ESM-2 embeddings trained on two different data regimes. When trained on general proteins, the loss landscape is relatively uniform across positions at each timestep, indicating balanced capacity allocation. When the same architecture is trained on antibody sequences, a qualitatively different pattern emerges. Loss concentrates in vertical bands corresponding to CDR positions, particularly CDR3 at positions roughly 100 and higher. These high-loss bands persist from $t = 1.0$ down to approximately $t = 0.5$, indicating that the model never achieves reliable denoising in these regions. Notably, the framework regions do not simply match the general-protein baseline. They fall *below* it, suggesting the model has become overconfident in these conserved regions. This dual pathology reflects the capacity misallocation predicted by the information density imbalance. The model undertrains on hypervariable CDRs while overfitting to the nearly deterministic frameworks.

To move beyond visual inspection, we compute two summary statistics. First, the standard deviation of per-position loss is 8.4×10^{-2} for antibodies versus 2.1×10^{-2} for general proteins, confirming that antibody training produces roughly twice the spatial variation in learning difficulty. Second, we correlate the per-position loss (averaged across timesteps) with the per-position entropy of the antibody sequences. The Pearson correlation is $r = 0.41$ ($p < 10^{-5}$), directly linking the information density imbalance in the data to the capacity misallocation in the model.

This locates the source of capacity misallocation in the latent representation. Two natural interventions follow: adapt the diffusion process to position-specific difficulty, or learn a representation that redistributes information evenly. We report process-level results (per-position loss reweighting, adaptive noise schedules) in Sec. E.5. Both rely on per-position statistics defined on a fixed positional coordinate system (here, IMGT alignment). Representation-level redistribution instead learns capacity allocation from data, and is the approach we pursue in Sec. 4.

4. Learning Redistributed Latent Representations

We address capacity misallocation by learning a latent representation where information density is redistributed by construction. Our approach (Fig. 3) employs an autoencoder: a compressor that maps variable-length sequences to fixed-dimension latents paired with a decompressor that inverts this mapping. The key mechanism relies on a hard capacity constraint. By fixing the latent dimension to $N < L$ where L is the typical sequence length, we force the model to allocate representation space efficiently. The compressor cannot waste capacity encoding redundant low-entropy framework positions—it must reallocate that capacity to the high-entropy CDR regions. We then train a diffusion model on these redistributed latents.

Training proceeds in two stages. First, we train the autoencoder end-to-end to compress and reconstruct sequences while learning a latent manifold suitable for diffusion (Sec. 4.2). Second, we freeze the trained autoencoder and train a standard Gaussian diffusion model on the compressed representations. At generation time, we sample from the diffusion model and decode the resulting latent through the frozen decompressor to obtain sequences.

4.1. Architecture

During **autoencoder** training, we first encode each antibody sequence $\mathbf{w} = (w_1, \ldots, w_L)$ with a frozen pretrained protein language model E_{PLM}, producing token-level representations $\mathbf{h} = E_{\text{PLM}}(\mathbf{w}) \in \mathbb{R}^{L \times d_{\text{enc}}}$, where d_{enc} is the encoder hidden dimension (e.g., $d_{\text{enc}} = 1280$ for ESM-2). These representations remain frozen throughout training and serve as the reconstruction target for our autoencoder.

We **compress** $\mathbf{h}$ into a fixed-length latent representation using a 12-layer cross-attention transformer (Alayrac et al., 2022). A set of N learnable latent queries aggregates information from the full token sequence into a compressed representation $\mathbf{z} \in \mathbb{R}^{N \times d}$. Full architectural details, including cross-attention mechanics, the decompression pathway, and token decoder training, appear in Sec. C.

We set $N = 128$ and $d = 16$ for all experiments. For typical

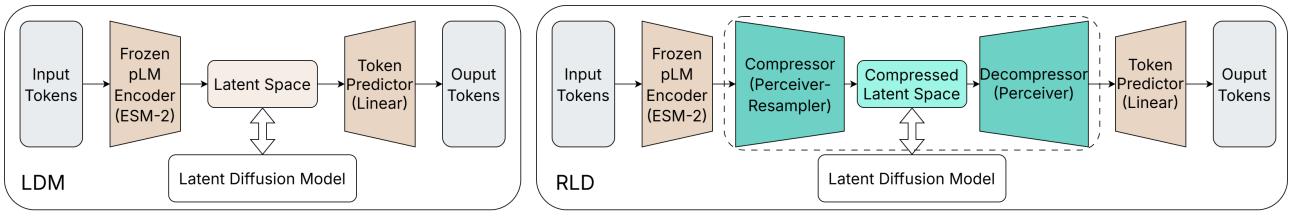


Figure 3. **Latent Diffusion vs. Redistributed Latent Diffusion.** *(Left)* LDM diffuses directly on frozen pLM hidden states. *(Right)* RLD inserts a learned compressor and decompressor, so diffusion operates on a compact fixed-length latent space whose information density is redistributed by the compression bottleneck.

antibody lengths of $L \approx 100$–140, $N = 128$ imposes a mild but consequential bottleneck along the sequence axis. Because $N \leq L$, the model cannot devote a separate latent slot to every input position, so it must pool predictable framework information and preserve capacity for the more variable CDR regions. This is the architectural source of redistribution, while the small feature dimension $d = 16$ keeps the latent space compact. Dimension ablations appear in Sec. E.2.

The **decompressor** mirrors the compressor with a fresh set of L learnable queries that attend to the latent $\mathbf{z}$ and expand it back into token-level representations $\hat{\mathbf{h}} \in \mathbb{R}^{L \times d_{\text{enc}}}$. The **token decoder** then projects each position of $\hat{\mathbf{h}}$ to amino-acid logits to reconstruct the sequence. Full details on the architecture appear in Sec. C. Training details for the autoencoder appear in Sec. 4.2.

Before applying diffusion, we **normalize** $\mathbf{z}$ to have zero mean and unit variance per dimension to stabilize diffusion training. We then train a Gaussian diffusion model on the normalized latents. At generation time, we sample latents from the trained diffusion model and decode them through the frozen decompressor and token decoder to obtain sequences. We defer the forward process, objective, and architectural details to Sec. C.3.

4.2. Training for Diffusable Latent Spaces

High reconstruction accuracy alone does not guarantee that the latent manifold is suitable for diffusion. An autoencoder trained solely to minimize reconstruction loss may learn a brittle geometry where latents cluster tightly around training examples with undefined behavior in the interpolation regions that diffusion must traverse during sampling. We augment the basic reconstruction objective with three training strategies that regularize the latent space for diffusability.

Our training employs **dual reconstruction losses** that operate at different semantic levels:

$$\mathcal{L}_{\text{AE}} = \mathcal{L}_{\text{CE}}(\mathbf{w}, \hat{\mathbf{w}}) + \mathcal{L}_{\text{MSE}}(\mathbf{h}, \hat{\mathbf{h}}), \tag{1}$$

where $\mathcal{L}_{\text{CE}}$ is token-level cross-entropy between original and reconstructed sequences and $\mathcal{L}_{\text{MSE}}$ is mean squared error between original and reconstructed encoder activations. The cross-entropy term ensures accurate sequence reconstruction, while the MSE term preserves semantic structure in the frozen PLM representations. Empirically, this avoids a failure mode in which the decompressor's final-layer normalization collapses, a common issue when training with cross-entropy alone (Gao et al., 2021).

We also apply **input augmentations** by perturbing encoder outputs $\mathbf{h}$ before compression, forcing the model to recover clean representations from corrupted inputs. With equal probability, we use either **span masking** or **Gaussian noise**. Span masking zeros out contiguous spans of length 10–30 tokens in $\mathbf{h}$. We mask spans rather than individual tokens because single positions are often recoverable from local context, especially in antibody framework regions with strong positional cues. This pushes the model toward more robust global representations rather than local copying. Gaussian noise applies $\mathbf{h}' = \delta\mathbf{h} + \sqrt{1-\delta^2}\boldsymbol{\epsilon}$ with $\delta = 0.7$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, introducing smooth perturbations across all positions. Together, these augmentations prevent simple feature memorization. To reconstruct accurately from corrupted inputs, the model must capture underlying sequence semantics. This robustness transfers directly to diffusion, where the decoder must remain stable when given imperfect latents produced during iterative denoising.

Finally, we employ **latent dropout** by zeroing out a fraction $p = 0.4$ of individual features within each latent vector $\mathbf{z}$ during training. Unlike input masking, which removes entire token representations, this acts at fine granularity within the compressed space. Randomly dropping dimensions forces information to be encoded redundantly, so that critical features cannot concentrate in a small subset of dimensions because any subset may be removed during training. The result is a distributed representation from which the decoder can reconstruct sequences from any sufficiently large fraction of latent features. This redundancy is important for diffusion, which must preserve semantic coherence while stochastically corrupting and restoring individual dimensions along the denoising trajectory.

4.3. Empirical Validation of Training Components

We validate that our training recipe produces the intended effects by measuring two properties: uniformity of capacity allocation and smoothness of the latent manifold. We

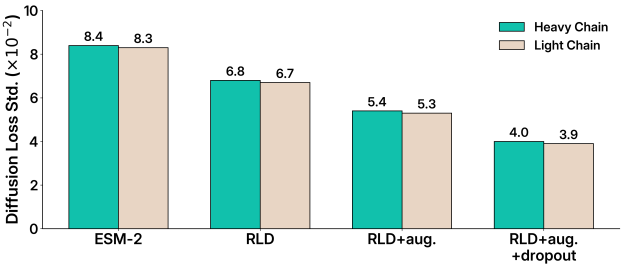


Figure 4. **Training components improve capacity uniformity.** Standard deviation of per-dimension diffusion loss across training configurations. Lower variance indicates more uniform capacity allocation. Results for human heavy chains; full metrics in Sec. E.3.

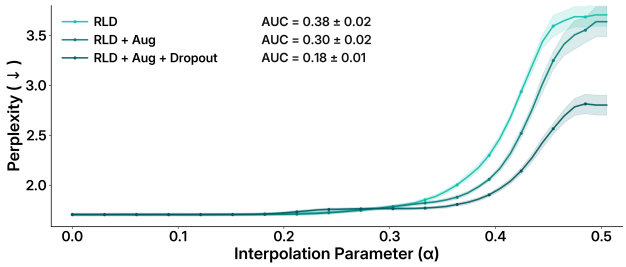


Figure 5. **Training components improve latent-space smoothness.** ProGen2 perplexity along linear interpolations between encoded sequence pairs, for three autoencoder configurations. Lower curves and lower AUC indicate smoother manifolds. Results for human heavy chains; full metrics in Sec. E.3.

compare four configurations: *(1)* raw ESM-2 representations used directly for diffusion; *(2)* an autoencoder trained with the reconstruction objectives (Eq. (1)); *(3)* the autoencoder with added input augmentations from Sec. 4.2; and *(4)* the full model with augmentations plus latent dropout. All measurements use human heavy and light chain sequences.

If **information density is successfully redistributed**, the diffusion model should allocate capacity uniformly across latent dimensions rather than concentrating it in easy framework regions. We use the standard deviation of per-dimension diffusion loss as a proxy for uniformity. Dimensions with higher loss receive more model capacity during training. Lower standard deviation indicates more uniform allocation. Tab. 9 shows progressive improvement. Raw ESM-2 representations exhibit high loss variance at 8.4×10^{-2} for heavy chains. The autoencoder provides moderate improvement to 6.8×10^{-2}. Adding input augmentations reduces variance substantially to 5.4×10^{-2}. Latent dropout further homogenizes the distribution to 4.0×10^{-2}. Fig. 4 visualizes this trend. Each training component contributes to more uniform capacity allocation.

A smooth latent manifold is essential for diffusion because the denoising trajectory must pass through intermediate regions where no training examples reside. We assess smoothness via linear interpolation between pairs of sequences. Given two encoded latents $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, we decode convex combinations $\mathbf{z}^{\alpha} = \alpha\mathbf{z}^{(1)} + (1-\alpha)\mathbf{z}^{(2)}$ for $\alpha \in [0, 1]$ and measure perplexity of the decoded sequence under ProGen2-base (Nijkamp et al., 2022). Mid-range α values probe regions unseen during training. A smooth manifold yields consistently low perplexity throughout the interpolation path. We report the area under the perplexity curve where lower values indicate smoother manifolds. Tab. 9 shows that the autoencoder achieves AUC of 0.38 for heavy chains. Adding input augmentations provides additional gains with AUC of 0.30. Latent dropout improves substantially to AUC of 0.18. Fig. 5 shows the full interpolation curves. Each training component independently reduces peak perplexity and stabilizes the trajectory.

We also validated the architectural choices of $N = 128$ and $d = 16$, testing combinations of $N \in \{16, 32, 64, 128, 256\}$ and $d \in \{16, 32, 64, 128, 256, 512, 1024\}$. The model exhibits stable performance across this range (Sec. E.2), confirming robustness to moderate variations in latent dimension.

The full training procedure transforms raw ESM representations with severe loss imbalance ($\sigma = 8.4 \times 10^{-2}$) into redistributed latents with substantially more uniform capacity allocation ($\sigma = 4.0 \times 10^{-2}$). This enables standard global noise schedules to provide consistent learnable signal across all latent dimensions, addressing the capacity misallocation identified in Sec. 3. We now evaluate whether these latent-space properties translate to generation quality on antibody sequences.

4.4. Comparison with Position-Aware Alternatives

Capacity misallocation can also be addressed within the diffusion process while leaving the representation fixed. We evaluate two such interventions on frozen ESM-2 latents: per-position loss reweighting by local entropy, and an adaptive per-position noise schedule that warps time per position to equalize accumulated loss profiles. The adaptive schedule reduces FD by 4–6× relative to the LDM baseline and recovers diversity. Loss reweighting raises diversity but increases FD on both chains. RLD reaches lower FD on heavy chains than the adaptive schedule and matches it on light.

Table 1. **Position-aware diffusion alternatives.** Comparison on human heavy (H) and light (L) chains. FD values scaled by 10^{-2}. Full results with standard deviations in Tab. 12.

	FD ↓ (10^{-2})		CD ↑	
Method	H	L	H	L
Dataset	0.36	0.25	0.971	0.839
LDM (ESM-2)	6.15	4.60	0.283	0.028
+ loss reweighting	12.85	11.56	0.566	0.746
+ adaptive schedule	1.47	0.75	**0.986**	**0.800**
RLD (ESM-2)	**0.57**	**0.61**	0.978	0.789

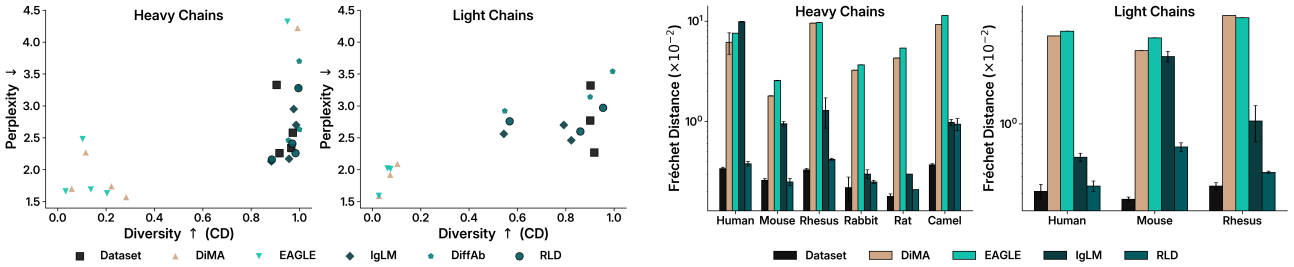


Figure 6. **Organism-conditioned generation.** *(Left)* Perplexity vs. Diversity (CD), one point per organism. Standard LDMs (DiMA, EAGLE) cluster in the low-CD, low-PPL region. RLD clusters with natural sequences. *(Right)* Fréchet distance across organisms and chain types. Standard LDMs show up to 23× higher FD than natural sequences (Black).

Both alternatives rely on per-position statistics defined on a fixed positional coordinate system (here, IMGT alignment), while RLD learns redistribution end-to-end. Full results and discussion appear in Sec. E.5.

5. Experiments

We evaluate redistributed latent diffusion (RLD) across multiple axes. After establishing our evaluation framework (§5.1), we show that redistributed latents resolve capacity misallocation across organisms and chain types (§5.2). To verify the improvement stems from the representation strategy rather than encoder choice, we replicate across five encoders spanning general protein models and antibody-specific models (§5.3). We then evaluate conditional generation: CDR infilling (§5.4), paired heavy-light chain generation (§5.5), and structure-conditioned generation (§5.6).

5.1. Evaluation Metrics

We evaluate using **Fréchet Distance (FD)** in ProtT5 embedding space, **ProGen2 perplexity (PPL)**, and **Clustering Density (CD)** at 95% sequence identity. We use the stringent 95% threshold because antibodies share a conserved immunoglobulin fold that renders the standard 50% threshold uninformative. PPL substantially below real-sequence values combined with low CD indicates mode collapse. We report mean ± standard deviation across five runs of 2,048 generated sequences. Full definitions and protocol details are in Sec. B.

5.2. Organism-Conditioned Generation

We evaluate RLD on generating antibody sequences conditioned on organism and chain type, testing whether redistributed latents resolve capacity misallocation across biologically diverse contexts.

Setup. We train RLD using ESM3 (Hayes et al., 2025) as the base encoder, which also supports our structure-conditioned experiments (Sec. 5.6). The latent has $N = 128$ and $d = 16$. The model conditions on organism (Human, Mouse, Rhesus, Rat, Rabbit, Camel) and chain type (Heavy/Light). Training details in Sec. D.

Baselines. Our baselines span four families: autoregressive models (IgLM (Shuai et al., 2023), p-IgGen (Turnbull et al., 2024)), latent diffusion on ESM-2 without redistribution (DiMA (Meshchaninov et al., 2025b), EAGLE (Cohen & Schneidman-Duhovny, 2023)), multinomial diffusion (Hoogeboom et al., 2021) as implemented in DiffAb (Luo et al., 2022), and Bayesian Flow Networks (AbBFN2 (Guloglu et al., 2025), IgCraft (Greenig et al., 2025)). AbBFN2, IgCraft, and p-IgGen are evaluated on human chains only. All other baselines are evaluated across all organisms.

Results. Figure 6 presents results across organisms and chain types. RLD approaches natural sequence statistics, reducing Fréchet distance to within 12% of the dataset on Human Heavy chains (0.0038 vs. 0.0034) and recovering clustering density to 0.970 (natural: 0.972). In contrast, DiMA uses identical architecture but standard pretrained ESM-2 latents, and exhibits the predicted consequence of capacity misallocation: Fréchet distance 18× higher than natural sequences, CD collapsed to 0.283, and perplexity 1.57 compared to 2.58 for natural sequences. The anomalously low perplexity indicates the model has become overconfident by memorizing conserved framework regions while collapsing CDR predictions to their mean (Fig. 6 (Left)).

This improvement generalizes across organisms. RLD reduces Fréchet distance by 7–23× relative to DiMA across all six species, with the largest gains on Rhesus (23×) and Rat (20×), organisms where pretrained encoder representations are likely weakest. Light chains show more severe baseline failure: DiMA's clustering density falls to 0.028 on Human Light chains versus 0.283 on Heavy chains. RLD recovers Light chain diversity to 0.860, approaching but not fully matching natural sequences (0.901). Nearest-neighbor identity to the training set matches that of held-out test sequences across organisms and chain types, ruling out memorization (Sec. F).

Among baselines, the autoregressive model IgLM maintains reasonable diversity without exhibiting capacity

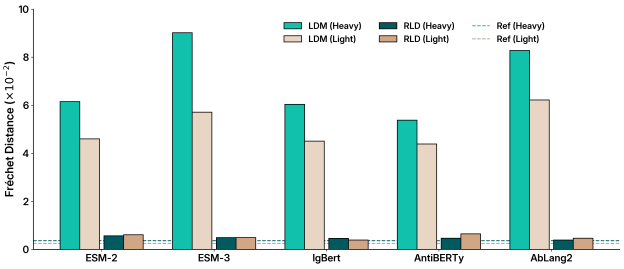


Figure 7. **Latent redistribution fixes capacity misallocation regardless of encoder architecture.** Fréchet Distance (FD, lower is better) for diffusion models trained on five encoders, comparing base latents (lighter bars) against RLD-redistributed latents (darker bar). Dashed lines indicate dataset reference.

misallocation, as expected since autoregressive generation lacks global noise schedules. RLD nonetheless achieves lower Fréchet distance than IgLM on most organisms. DiffAb produces diverse but distributionally mismatched sequences (high CD, high FD), a complementary failure mode. AbBFN2 and IgCraft achieve strong diversity but 4–7× higher Fréchet distance than RLD on Human chains.

These results use ESM3 as the base encoder. To verify the improvement stems from the representation strategy rather than encoder choice, we next replicate across five encoder architectures.

5.3. Encoder Generalization

A natural question is whether the gains in Sec. 5.2 stem from redistribution or from idiosyncrasies of the ESM3 encoder. We replicate across five encoder architectures spanning general protein models and antibody-specific models.

Setup. We evaluate two general protein language models (ESM-2 (Lin et al., 2023), ESM3 (Hayes et al., 2025)) and three antibody-specific encoders (IgBert (Kenlay et al., 2024), AntiBERTy (Ruffolo et al., 2021), AbLang2 (Olsen et al., 2024)). For each encoder, we train two diffusion models, one with standard pretrained latents and one with redistributed latents. All models use identical training protocols and are evaluated on Human heavy and light chains. Full metrics and per-organism breakdowns appear in Sec. E.1.

Results. Fig. 7 summarizes the effect of redistribution across encoders. Regardless of pretraining domain, every encoder exhibits the same failure mode when used with standard latents: FD values of 0.04–0.09, approximately an order of magnitude above the dataset distribution. Upon redistribution via RLD, FD drops to 0.004–0.006 across all encoders, approaching the reference (Tab. 7 reports corresponding shifts in PPL and CD).

One might expect antibody-specific encoders to have already learned representations suitable for antibody generation. Yet IgBert, AntiBERTy, and AbLang2 exhibit the same capacity misallocation as general protein models.

Table 2. **Variable-length CDR3 infilling.** Edit distance to reference, sequence divergence between samples, and ProGen2 perplexity on human heavy (H) and light (L) chains.

	Edit dist. ↓		Seq. div. ↑		PPL ↓	
Model	H	L	H	L	H	L
Eval set	0.00	0.00	0.00	0.00	2.59	2.89
RLD (Ours)	**0.74**	**0.39**	0.83	0.60	2.59	**2.87**
AbBFN2	0.79	0.40	**0.84**	0.62	2.59	**2.87**
IgLM	0.75	0.42	0.82	**0.67**	**2.55**	2.89

The failure is not due to lack of domain knowledge but to the incompatibility between heterogeneous latent spaces and homogeneous noise schedules diagnosed in Sec. 3.These results confirm that capacity misallocation is a *representation* problem, not an architecture problem. The proposed fix, learning redistributed representations, is encoder-agnostic.

5.4. CDR Infilling

CDR infilling tests whether the model has learned the joint distribution over framework and hypervariable regions. We focus on variable-length CDR3 infilling, where the model regenerates a masked CDR3 conditioned on the rest of the sequence and predicts the length of the inserted region. This setting reflects practical design workflows, where the target loop length is generally not known in advance.

Setup. For each masked sequence we generate $K = 100$ samples and report three metrics: normalized Levenshtein distance to the reference CDR3 (fidelity), mean pairwise Levenshtein distance between samples (sample-level diversity), and ProGen2 perplexity of the completed sequence (naturalness). We compare against AbBFN2 (Guloglu et al., 2025) and IgLM (Shuai et al., 2023), which natively handle variable-length CDR3 prediction.

Results. Table 2 summarizes results on human heavy and light chains. RLD is competitive with the strongest baseline on every metric and chain. Edit distance is lowest for RLD on both chains, sample divergence is within 0.01–0.07 of the best baseline, and perplexity matches reference statistics closely. The same architecture also supports joint infilling of all three CDRs at once, a capability not found in most antibody models. We also report fixed-length single-span and multi-span results in Sec. G, consistent with the variable-length setting above.

5.5. Heavy-Light Chain Pairing

Functional antibodies require coordinated heavy (VH) and light (VL) chains that form stable complexes. A practical model must produce pairs with the structural and sequence complementarity of natural antibodies (Guo et al., 2025).

Setup. We evaluate conditional paired generation, where the model generates one chain conditioned on its partner chain and organism label. We assess pairing quality through four metrics. *ImmunoMatch (IM-Score)* quantifies heavy-

Table 3. **Paired (VH:VL) generation results.** True Pairs are sequences from the test set. False Pairs are 100 random heavy-light pairings drawn from training.

Model	IM-Score ↑		|ΔPPL| ↓		CD ↑		Edit dist. ↑	
	VH	VL	VH	VL	VH	VL	VH	VL
True Pairs	0.73	0.75	0.00	0.00	—	—	0.00	0.00
False Pairs	—	—	0.04	0.19	1.00	0.97	0.44	0.43
RLD (Ours)	**0.71**	**0.74**	**0.03**	0.16	0.99	0.63	0.43	0.40
IgCraft	0.56	0.56	0.53	0.51	0.47	0.10	0.32	0.22
p-IgGen	0.70	**0.74**	0.29	**0.03**	**1.00**	**0.66**	**0.44**	**0.42**
pAbT5	0.61	0.64	0.48	0.49	0.98	0.38	0.42	0.41

Table 4. **Inverse-folding.** Global and CDR3 RMSD (Å) between IgFold-refolded generated sequences and reference structures for heavy (H) and light (L) chains.

Model	Global RMSD ↓		CDR3 RMSD ↓	
	H	L	H	L
RLD (Ours)	0.839	**0.605**	1.234	**0.710**
AntiFold	**0.797**	0.609	**1.130**	0.733
AbMPNN	1.353	0.704	1.638	0.829
ESM3	1.422	0.768	1.704	0.916
ProteinMPNN	2.245	0.803	3.864	0.976
ESM-IF1	3.629	1.146	5.071	1.436

light compatibility from learned features of natural pairings (Sec. B.7). |ΔPPL| is the absolute deviation in perplexity from real antibody sequences (Sec. B.2). *Clustering Density (CD)* measures output diversity. Normalized edit distance from training data measures novelty. Dataset and training details in Secs. A.2 and D.3.

Baselines. We compare against three paired antibody generation models, IgCraft (Greenig et al., 2025), p-IgGen (Turnbull et al., 2024), and pAbT5 (Chu & Wei, 2023), plus two reference conditions: *TP* (True Pairs from the test set) and *FP* (False Pairs created by randomly matching training sequences).

Results. Table 3 presents paired generation results. RLD achieves IM-Scores (VH: 0.71, VL: 0.74) approaching real antibody pairs (VH: 0.73, VL: 0.75) and matching the strongest baseline p-IgGen (VH: 0.70, VL: 0.74). It also produces natural sequences, with perplexity deviations (|ΔPPL|: 0.03, 0.16) substantially lower than all baselines on VH and competitive with the best baseline on VL. RLD maintains high diversity (CD: 0.99, 0.63), avoiding the mode collapse exhibited by IgCraft (CD: 0.47, 0.10) and matching p-IgGen and pAbT5 on heavy chains. Edit distances (VH: 0.43, VL: 0.40) confirm that RLD generates novel pairs rather than memorizing training examples.

These results show that redistributed latents enable coherent multi-chain generation, capturing the joint distribution over heavy-light pairs.

5.6. Structure-Conditioned Generation

We evaluate whether RLD's redistributed latent space supports structure conditioning, using inverse folding as the benchmark task.

Setup. We finetune the pretrained RLD model on antibody backbones. Conditioning is supplied as discrete structure tokens from the ESM3 structure VQ-VAE, encoded by a small auxiliary transformer and cross-attended into each score-estimator block (Sec. D.4). Training uses the antibody structure dataset of AntiFold (Høie et al., 2025) with the same train/test split as AntiFold and AbMPNN (Sec. A.3). We compare against antibody-specific inverse folding (AntiFold, AbMPNN (Dreyer et al., 2023)), general-protein inverse folding (ESM-IF1 (Hsu et al., 2022), ProteinMPNN (Dauparas et al., 2022)), and the structure-conditioned mode of ESM3 (Hayes et al., 2025). We refold each generated sequence with IgFold (Ruffolo et al., 2023) and report Global RMSD (Kabsch over all shared Cα after IMGT numbering) and CDR3 RMSD (Kabsch over framework Cα, RMSD measured on CDR3 positions). Test-set construction is detailed in Sec. H.1.

Results. Table 4 shows that RLD reaches inverse-folding performance comparable to AntiFold, the strongest specialized antibody baseline. The differences between RLD and AntiFold are around 0.1 Å or less across all four metrics. The remaining baselines trail by larger margins, with general-protein methods (ESM-IF1, ProteinMPNN) showing several-fold higher Global RMSD on light chains. Per-region amino-acid recovery follows the same pattern. RLD achieves the highest AAR on 11 of 12 region–chain combinations, with AntiFold leading only on heavy CDR3 (Tab. 19). These results extend RLD's conditional-generation capabilities from sequence inputs (Secs. 5.4 and 5.5) to 3D structure, using the same finetune-with-cross-attention recipe as the other conditional tasks.

6. Conclusion

We identified capacity misallocation as the failure mode that prevents standard latent diffusion from learning antibody sequence distributions, and traced it to the entropy gap between framework and CDR regions that pretrained encoders preserve in their latent space. Addressing this in the representation rather than in the diffusion process, we learned a compressed latent space that redistributes information across positions. Standard Gaussian diffusion on these latents matches natural sequence statistics across six organisms and five encoder architectures, and supports CDR infilling, paired-chain generation, and structure-conditioned generation within a single training recipe.

Acknowledgements

This research was supported in part through computational resources of HPC facilities at HSE University. The work was supported by the grant for research centers in the field

of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4E0002 and the agreement with HSE University №139-15-2025-009.

References

Abanades, B., Wong, W. K., Boyles, F., Georges, G., Bujotzek, A., and Deane, C. M. ImmuneBuilder: Deep-Learning models for predicting the structures of immune proteins. *Communications Biology*, 6(1): 575, May 2023. ISSN 2399-3642. doi: 10.1038/s42003-023-04927-7. URL https://www.nature.com/articles/s42003-023-04927-7. 13

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. 3

Atkinson, T., Barrett, T. D., Cameron, S., Guloglu, B., Greenig, M., Robinson, L., Graves, A., Copoiu, L., and Laterre, A. Protein sequence modelling with bayesian flow networks. *Nature Communications*, 16, 2024. 1

Bowle, A. M.-J. S. M. A. S. E. B. M. O. M. A. A., Bateman, A., Martin, M. J., Orchard, S., Magrane, M., Adesina, A. Y., Ahmad, S., Bowler-Barnett, E. H., Bye-A-Jee, H., Carpentier, D., Denny, P., Fan, J., Garmiri, P., da Costa Gonzales, L. J., Hussein, A., Ignatchenko, A., Insana, G., Ishtiaq, R., Joshi, V., Jyothi, D., Kandasaamy, S., Lock, A., Luciani, A., Luo, J., Lussi, Y., Marin, J. S. M., Raposo, P., Rice, D., Santos, R., Speretta, E., Stephenson, J. L., Totoo, P., Tyagi, N., Urakova, N., Vasudev, P., Warner, K., Wijerathne, S., Yu, C. W.-H., Zaru, R., Bridge, A. J., Aimo, L., Argoud-Puy, G., Auchincloss, A. H., Axelsen, K. B., Bansal, P., Baratin, D., Neto, T. M. B., Blatter, M.-C., Bolleman, J. T., Boutet, E., Breuza, L., Gil, B. C., Casals-Casas, C., Echioukh, K. C., Coudert, E., Cuche, B., de Castro, E., Estreicher, A., Famiglietti, M. L., Feuermann, M., Gasteiger, E., Gaudet, P., Gehant, S., Gerritsen, V. B., Gos, A., Gruaz, N., Hulo, C., Hyka-Nouspikel, N., Jungo, F., Kerhornou, A., Mercier, P. L., Lieberherr, D., Masson, P., Morgat, A., Paesano, S., Pedruzzi, I., Pilbout, S., Pourcel, L., Poux, S., Pozzato, M., Pruess, M., Redaschi, N., Rivoire, C., Sigrist, C. J. A., Sonesson, K., Sundaram, S., Sveshnikova, A., Wu, C. H., Arighi, C. N., Chen, C., Chen, Y., Huang, H., Laiho, K., Lehvaslaiho, M., McGarvey, P. B., Natale, D. A., Ross, K., Vinayaka, C. R., Wang, Y., and Zhang, J. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53:D609 – D617, 2024. 14

Briney, B. S., Inderbitzin, A., Joyce, C., and Burton, D. R. Commonality despite exceptional diversity in the baseline human antibody repertoire. *Nature*, 566:393 – 397, 2018. 1, 2

Chu, S. K. S. and Wei, K. Y. Generative antibody design for complementary chain pairing sequences through encoder-decoder language model. *ArXiv*, abs/2301.02748, 2023. 8

Cohen, T. and Schneidman-Duhovny, D. Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. 6

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Courbet, A., de Haas, R. J., Bethel, N., Leung, P. J. Y., Huddy, T. F., Pellock, S., Tischer, D., Chan, F., Koepnick, B., Nguyen, H., Kang, A., Sankaran, B., Bera, A. K., King, N. P., and Baker, D. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022. doi: 10.1126/science.add2187. URL https://www.science.org/doi/abs/10.1126/science.add2187. 8

Dieleman, S., Sartran, L., Roshannai, A., Savinov, N., Ganin, Y., Richemond, P. H., Doucet, A., Strudel, R., Dyer, C., Durkan, C., Hawthorne, C., Leblond, R., Grathwohl, W., and Adler, J. Continuous diffusion for categorical data. *ArXiv*, abs/2211.15089, 2022. 1, 24

Dreyer, F. A., Cutting, D., Schneider, C., Kenlay, H., and Deane, C. M. Inverse folding for antibody sequence design using deep learning, 2023. URL https://arxiv.org/abs/2310.19513. 8, 13

Dunbar, J. and Deane, C. M. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32:298 – 300, 2015. 14

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. Sabdab: the structural antibody database. *Nucleic Acids Research*, 42 (D1):D1140–D1146, 01 2014. ISSN 0305-1048. doi: 10.1093/nar/gkt1043. URL https://doi.org/10.1093/nar/gkt1043. 13

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Fehér, T. B., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020. 13

Frey, N. C., Berenberg, D., Zadorozhny, K., Kleinhenz, J., Lafrance-Vanasse, J., Hotzel, I., Wu, Y., Ra, S., Bonneau, R., Cho, K., Loukas, A., Gligorijević, V., and Saremi, S. Protein discovery with discrete walk-jump sampling. *ArXiv*, abs/2306.12360, 2023. 1

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821, 2021. 4

Greenig, M., Zhao, H., Radenkovic, V., Ramon, A., and Sormanni, P. Igcraft: A versatile sequence generation framework for antibody discovery and engineering. *ArXiv*, abs/2503.19821, 2025. 6, 8

Greenshields-Watson, A., Agarwal, P., Robinson, S. A., Williams, B. H., Gordon, G. L., Capel, H. L., Li, Y., Spoendlin, F. C., Aguilar-Sanjuan, B., Boyles, F., and Deane, C. M. Anarcii: A generalised language model for antigen receptor numbering. *bioRxiv*, 2025. 13

Guloglu, B., Bragança, M., Graves, A., Cameron, S., Atkinson, T., Copoiu, L., Laterre, A., and Barrett, T. D. Abbfn2: A flexible antibody foundation model based on bayesian flow networks. *bioRxiv*, 2025. 6, 7

Guo, D., Dunn-Walters, D. K., Fraternali, F., and Ng, J. C. F. Immunomatch learns and predicts cognate pairing of heavy and light immunoglobulin chains. *Nature Methods*, 23:106 – 117, 2025. 7, 15

Hayes, T., Rao, R., Akin, H., Sofroniew, N. J., Oktay, D., Lin, Z., Verkuil, R., Tran, V. Q., Deaton, J., Wiggert, M., Badkundri, R., Shafkat, I., Gong, J., Derry, A., Molina, R. S., Thomas, N., Khan, Y. A., Mishra, C., Kim, C., Bartie, L. J., Nemeth, M., Hsu, P. D., Sercu, T., Candido, S., and Rives, A. Simulating 500 million years of evolution with a language model. *Science*, pp. eads0018, 2025. 6, 7, 8

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 18

Hoogeboom, E., Nielsen, D., Jaini, P., Forr'e, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Neural Information Processing Systems*, 2021. 6

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/09/06/2022.04.10.487779. 8

Høie, M. H., Hummer, A. M., Olsen, T. H., Aguilar-Sanjuan, B., Nielsen, M., and Deane, C. M. Antifold: improved structure-based antibody design using inverse folding. *Bioinformatics Advances*, 5(1):vbae202, 01 2025. ISSN 2635-0041. doi: 10.1093/bioadv/vbae202. URL https://doi.org/10.1093/bioadv/vbae202. 8, 13

Kenlay, H., Dreyer, F. A., Kovaltsuk, A., Miketa, D., Pires, D. E. V., and Deane, C. M. Large scale paired antibody language models. *PLOS Computational Biology*, 20, 2024. 7

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *ArXiv*, abs/2107.00630, 2021. 2

Lee, S., Lee, K., and Park, T. Ant: Adaptive noise schedule for time series diffusion models. *ArXiv*, abs/2410.14488, 2024. 1

Lefranc, M.-P., Pommié, C., Ruiz, M., Giudicelli, V., Foulquier, E., Truong, L. D., Thouvenin-Contet, V., and Lefranc, G. Imgt unique numbering for immunoglobulin and t cell receptor variable domains and ig superfamily v-like domains. *Developmental and comparative immunology*, 27 1:55–77, 2003. 2, 13, 14

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., and Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. 2, 7

Luo, S., Su, Y., Peng, X., Wang, S., Peng, J., and Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 9754–9767. Curran Associates, Inc., 2022. 6

Meshchaninov, V., Chimbulatov, E., Shabalin, A., Abramov, A., and Vetrov, D. Compressed and smooth latent space for text diffusion modeling. *ArXiv*, abs/2506.21170, 2025a. 1

Meshchaninov, V., Strashnov, P. V., Shevtsov, A., Nikolaev, F., Ivanisenko, N. V., Kardymon, O. L., and Vetrov, D. Diffusion on language model encodings for protein sequence generation. In *International Conference on Machine Learning*, 2025b. 1, 6, 18

Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N. V., and Madani, A. Progen2: Exploring the boundaries of protein language models. *Cell systems*, 2022. 5, 13

Olsen, T. H., Boyles, F., and Deane, C. M. Observed antibody space: A diverse database of cleaned, annotated, and translated unpaired and paired antibody sequences. *Protein Science : A Publication of the Protein Society*, 31:141 – 146, 2021. 13

Olsen, T. H., Moal, I. H., and Deane, C. M. Ablang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2, 2022. 13

Olsen, T. H., Moal, I. H., and Deane, C. M. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40, 2024. 7

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2021. 1, 2

Ruffolo, J. A., Gray, J. J., and Sulam, J. Deciphering antibody affinity maturation with language models and weakly supervised learning. *ArXiv*, abs/2112.07782, 2021. 7

Ruffolo, J. A., Chu, L.-S., Mahajan, S. P., and Gray, J. J. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature Communications*, 14(1):2389, April 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38063-x. URL https://www.nature.com/articles/s41467-023-38063-x. 8, 30

Sahoo, S. S., Gokaslan, A., Sa, C. D., and Kuleshov, V. Diffusion models with learned adaptive noise. *ArXiv*, abs/2312.13236, 2023. 1

Shah, H., Tamuly, K., Raghunathan, A., Jain, P., and Netrapalli, P. The pitfalls of simplicity bias in neural networks. *ArXiv*, abs/2006.07710, 2020. 2

Shuai, R. W., Ruffolo, J. A., and Gray, J. J. Iglm: Infilling language modeling for antibody sequence design. *Cell systems*, 2023. 6, 7, 13

Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020. 2

Steinegger, M. and Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35:1026–1028, 2017. 14

Strashnov, P., Shevtsov, A., Meshchaninov, V., Kardymon, O., and Vetrov, D. Geommotif: A benchmark for arbitrary geometric preservation in protein generation. *bioRxiv*, 2026. doi: 10.64898/2026.04.18.719378. URL https://www.biorxiv.org/content/early/2026/04/22/2026.04.18.719378. 30

Tonegawa, S. Somatic generation of antibody diversity. *Nature*, 302:575–581, 1983. 1

Turnbull, O. M., Oglic, D., Croasdale-Wood, R., and Deane, C. M. p-iggen: a paired antibody generative language model. *Bioinformatics*, 40, 2024. 6, 8

Weitzner, B. D., Dunbrack, R. L., and Gray, J. J. The origin of cdr h3 structural diversity. *Structure*, 23 2:302–11, 2015. 1

Yim, J., Trippe, B. L., Bortoli, V. D., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se(3) diffusion model with application to protein backbone generation. *ArXiv*, abs/2302.02277, 2023. 14

Appendix

Contents

A Dataset **13**
- A.1 Unpaired Dataset . . . 13
- A.2 Paired Dataset . . . 13
- A.3 Structure Dataset . . . 13

B Evaluation metrics **13**
- B.1 Distributional Similarity . . . 13
- B.2 Sequence Plausibility . . . 13
- B.3 Diversity . . . 14
- B.4 Diversity Score for CDR Infilling . . . 14
- B.5 Amino Acid Recovery (AAR) . . . 14
- B.6 Edit Distance. . . . 15
- B.7 IM-score. . . . 15
- B.8 Structure-Based Metrics . . . 15

C Architecture Details **17**
- C.1 Autoencoder Architecture . . . 17
- C.2 Length Handling and Normalization . . . 17
- C.3 Diffusion Denoiser . . . 18

D Training Details **18**
- D.1 Pretraining . . . 18
- D.2 Finetuning on infilling task . . . 18
- D.3 Finetuning on pairing task . . . 18
- D.4 Finetuning on structure conditioning task . . . 19
- D.5 Compute Resources . . . 19

E Additional Experiments **19**
- E.1 RLD Generalization to Different Encoder Models . . . 19
- E.2 Impact of Autoencoder Dimensions . . . 21
- E.3 Autoencoder Ablation . . . 23
- E.4 Latent Space Smoothness Evaluation . . . 23
- E.5 Per-Position Loss Reweighting and Noise Schedules . . . 24

F Organism-Conditioned Generation **26**

G Sequence Infilling **27**
- G.1 Variable-Length CDR3 Infilling . . . 27
- G.2 Single-Span Fixed-Length CDR3 Infilling . . . 28
- G.3 Multi-Span Fixed-Length All-CDR Infilling . . . 29
- G.4 Implications . . . 29

H Structure-Conditioned Generation **30**
- H.1 Test-Set Construction . . . 30
- H.2 RMSD Results . . . 30
- H.3 Per-Region Amino-Acid Recovery . . . 30
- H.4 Broader Impacts . . . 31

A. Dataset

A.1. Unpaired Dataset

We use the Observed Antibody Space (OAS) database (Olsen et al., 2021) as our data source, following the preprocessing approach of IgLM (Shuai et al., 2023). IgLM applied clustering with 95% sequence identity threshold to increase diversity. We then apply additional quality filters to remove low-quality sequences.

Following AbLang (Olsen et al., 2022), we note that over 40% of OAS sequences are missing the first 15 amino acids. To address this issue, we use ANARCII (Greenshields-Watson et al., 2025) to number all sequences using IMGT (Lefranc et al., 2003) scheme and remove sequences with numbering gaps at the first framework position for both heavy and light chains. We also filter light chain sequences shorter than 90 residues to remove an abnormal mode in the length distribution. Our filtering retains 203,968,933 training sequences and 11,819,794 test sequences across 6 organism labels (human, mouse, rat, rabbit, camel, rhesus) and 2 chain types (heavy, light).

A.2. Paired Dataset

For heavy-light chain pairing experiments, we use the paired human sequences from OAS (Olsen et al., 2021). We restrict to human sequences due to low population counts for other organisms in the paired OAS dataset. We apply the same quality filters as the unpaired dataset, though these filters remove fewer sequences in the paired setting. The final paired dataset contains 2,505,919 training sequences and 24,345 test sequences.

A.3. Structure Dataset

For structure-conditioned generation we use the antibody structure dataset assembled by AntiFold (Høie et al., 2025). The dataset combines two sources: 2,074 experimentally determined paired antibody structures from SAbDab (Dunbar et al., 2014), and 147,458 antibodies modelled from OAS (Olsen et al., 2021) sequences using ABodyBuilder2 (Abanades et al., 2023). With both heavy and light chains this yields roughly 300,000 chain structures. We adopt the same train/test split as both AntiFold and AbMPNN (Dreyer et al., 2023), which makes our results directly comparable to theirs.

B. Evaluation metrics

B.1. Distributional Similarity

Fréchet ProtT5 Distance (FD). We measure distributional similarity between generated and real sequences using the Fréchet distance (equivalently, the 2-Wasserstein distance) computed in the embedding space of ProtT5 (Elnaggar et al., 2020). Given two sets of sequences embedded as samples from multivariate Gaussians $X_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $X_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$, the Fréchet distance is:

$$d(X_1, X_2)^2 = ||\mu_1 - \mu_2||^2 + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2\sqrt{\Sigma_1 \Sigma_2}\right) \tag{2}$$

We obtain fixed-size embeddings by applying global average pooling along the sequence length dimension. FD equals zero when generated and real distributions match perfectly, with larger values indicating greater distributional divergence.

B.2. Sequence Plausibility

ProGen2 Perplexity (PPL). We quantify sequence plausibility using perplexity under ProGen2-base (Nijkamp et al., 2022), a 764M-parameter protein language model. For a sequence $S = (s_1, \ldots, s_{|S|})$:

$$\mathrm{PPL}(S) = \exp\left\{-\frac{1}{|S|}\sum_{i=1}^{|S|} \log p(s_i \mid s_{<i}; \Theta_{\text{ProGen2}})\right\} \tag{3}$$

Lower perplexity indicates sequences that better conform to natural protein statistics. However, perplexity alone can be misleading: a model that repeatedly generates a small set of high-likelihood sequences will achieve low perplexity while failing to capture distributional diversity. We therefore interpret PPL jointly with diversity metrics, treating perplexity substantially *below* real-sequence values as a potential indicator of mode collapse.

Perplexity Difference (ΔPPL). For some experiments, we report perplexity difference rather than raw perplexity. This metric measures deviation from a reference distribution and is particularly useful for conditional generation tasks where we

wish to assess whether generated sequences match the statistical properties of natural sequences in the same condition. We compute:

$$\Delta\text{PPL} = |\text{PPL}_{\text{generated}} - \text{PPL}_{\text{reference}}| \tag{4}$$

where $\text{PPL}_{\text{reference}}$ is the mean perplexity of natural sequences from the corresponding test set (e.g., same organism, chain type, or infilling context). Lower ΔPPL indicates that generated sequences exhibit perplexity closer to natural sequences. This normalization is particularly valuable when comparing across different experimental settings where absolute perplexity values may vary due to sequence length or composition differences.

B.3. Diversity

Clustering Density (CD). We assess diversity using clustering density, defined as the ratio of sequence clusters to total generated sequences after clustering at a given sequence identity threshold. CD ranges from 0 to 1, where CD = 1 indicates all sequences are distinct (each forms its own cluster) and lower values indicate redundant outputs.

Clustering thresholds are domain-dependent. For general proteins, $\text{CD}_{0.5}$ at 50% sequence identity is standard (Bowle et al., 2024), analogous to the TM-score threshold of 0.5 used in structure generation (Yim et al., 2023). However, antibody sequences exhibit substantially lower natural diversity than general proteins due to their shared immunoglobulin fold and conserved framework regions. At the 50% threshold, even biologically distinct antibodies cluster together, rendering $\text{CD}_{0.5}$ uninformative. We therefore report $\text{CD}_{0.95}$ at 95% sequence identity, which captures meaningful variation within the antibody sequence space and identifies near-duplicate sequences that would indicate mode collapse.

We perform clustering using MMseqs2 (Steinegger & Söding, 2017) with parameters: coverage = 0.8, cov-mode = 0, cluster-mode = 1.

B.4. Diversity Score for CDR Infilling

For CDR infilling experiments, we compute diversity across multiple samples generated for each masked sequence to assess whether models produce varied or repetitive predictions. The diversity metric operates at the task level, requiring multiple samples per masked sequence.

For each task, we generate $K = 100$ samples and extract the filled CDR regions. We compute three components: (1) **Unique fraction**, the proportion of distinct generated fills among all K samples; (2) **Mean pairwise identity**, the average sequence identity across all pairs of unique fills, where higher values indicate less diversity; (3) **Per-position entropy**, measuring amino acid variability at each position. The overall diversity score combines these as diversity $=$ $(\text{unique fraction} + (1 - \text{mean pairwise identity}))/2$, where higher values indicate more diverse predictions.

Deterministic decoding strategies such as argmax sampling yield zero diversity by design, as they always produce identical outputs for the same input. In Figure 11, AbLang and AbLang2 models have no visible markers due to zero diversity scores. For stochastic models, diversity score is visualized via marker size, where larger markers indicate more varied predictions across samples.

B.5. Amino Acid Recovery (AAR)

Amino Acid Recovery measures the fraction of positions where a generated sequence exactly matches the ground truth. For infilling tasks, this quantifies how faithfully a model reconstructs masked regions given surrounding context.

Challenge with Variable-Length Sequences. Standard AAR computation assumes fixed-length sequences where position indices directly correspond between generated and reference sequences. However, antibody generation models may produce sequences of varying lengths. Autoregressive models like IgLM generate sequences token-by-token without length constraints, potentially producing insertions or deletions relative to the reference sequence. Computing AAR on such sequences requires alignment to establish position correspondence.

Alignment-Based AAR. We compute AAR using structure-aware alignment through ANARCI (Dunbar & Deane, 2015), a tool that assigns standardized IMGT numbering (Lefranc et al., 2003) to antibody sequences. IMGT numbering is a universal coordinate system for antibody sequences that accounts for insertions and deletions while preserving structural correspondence across diverse antibodies.

Our procedure is as follows:

1. Number both the reference sequence and generated sequence using ANARCI with IMGT scheme.
2. For each IMGT position that exists in both sequences, check if the amino acids match.
3. Compute AAR as the fraction of matching positions over all positions present in both sequences.
4. Positions where both sequences have gaps are excluded from the calculation.

Formally, let r denote the reference sequence and g denote the generated sequence after IMGT numbering. Let P_r and P_g denote the sets of IMGT positions present in r and g respectively (excluding gap positions). Then:

$$\text{AAR} = \frac{|\{p \in P_r \cap P_g : r_p = g_p\}|}{|P_r \cap P_g|}$$

where r_p and g_p denote the amino acids at IMGT position p in the reference and generated sequences.

Equivalence to Standard AAR. For models that generate fixed-length sequences matching the reference length, IMGT numbering produces identical position assignments, and alignment-based AAR reduces to standard position-wise AAR. This ensures our metric is comparable across both fixed-length models (e.g. AbBFN2, ESM3) and variable-length models (e.g. IgLM).

Interpretation. AAR measures exact amino acid match and thus represents a strict notion of fidelity. A value of 0.7 indicates that 70% of structurally corresponding positions have identical amino acids. However, we note that AAR does not account for biochemical similarity: a substitution from leucine to isoleucine (both hydrophobic, structurally similar) is penalized equally to a substitution from leucine to aspartic acid (hydrophobic to charged). Additionally, multiple CDR sequences may be functionally compatible with the same framework, so lower AAR does not necessarily indicate functional failure. We therefore interpret AAR in conjunction with perplexity and diversity metrics to assess overall infilling quality.

B.6. Edit Distance.

Edit Distance measures sequence-level similarity between generated and reference sequences. We compute the mean normalized Levenshtein distance across all generated sequences, where normalization divides by the maximum sequence length in each pair. Lower values indicate generated sequences are closer to the reference distribution in terms of exact character-level matches.

B.7. IM-score.

To validate heavy-light chain pairing compatibility, we use the ImmunoMatch model (Guo et al., 2025), which was trained on paired heavy and light chain sequences from human B cells to distinguish cognate from random chain pairs. Specifically, we employ the ImmunoMatch variant trained without explicit light chain type specification (i.e., without distinguishing between κ and λ chains).

Rather than using raw ImmunoMatch scores directly, we construct a calibrated metric called IM-score as follows. For each validation task, we first compute raw ImmunoMatch scores for 100 randomly shuffled (false) heavy-light pairs from the same dataset. We then express each true pair's raw score as a percentile rank within this empirical negative distribution. This percentile ranking approach provides a normalized metric that is robust to potential score scale variations across different sequence sets and ensures interpretability: higher IM-scores indicate greater compatibility relative to random pairings from the same population.

B.8. Structure-Based Metrics

Antibody sequences share a conserved immunoglobulin fold. As a consequence, structure-prediction-based metrics applied to generated sequences saturate near the values seen for natural antibodies and provide little discriminative signal between methods. We illustrate this with two standard structural-fidelity metrics: ESMFold pLDDT (per-residue confidence of a

predicted structure) and IgFold pRMSD (predicted backbone RMSD). For each metric, we generate 512 sequences with each model and repeat four times.

Table 5. **Human heavy-chain structural fidelity.** ESMFold pLDDT (higher is more confident) and IgFold pRMSD (lower is closer to a confidently-predicted structure). Mean and standard deviation across four runs of 512 generated sequences.

Model	ESMFold pLDDT	IgFold pRMSD
Test set	83.6 ± 0.2	0.38 ± 0.01
LDM (ESM-2)	85.1 ± 0.1	0.31 ± 0.01
RLD (ESM-2)	83.9 ± 0.3	0.37 ± 0.01
LDM (ESM3)	85.6 ± 0.2	0.29 ± 0.00
RLD (ESM3)	83.9 ± 0.2	0.38 ± 0.00

Table 6. **Human light-chain structural fidelity.** ESMFold pLDDT (higher is more confident) and IgFold pRMSD (lower is closer to a confidently-predicted structure). Mean and standard deviation across four runs of 512 generated sequences.

Model	ESMFold pLDDT	IgFold pRMSD
Test set	87.4 ± 0.2	0.34 ± 0.00
LDM (ESM-2)	88.4 ± 0.2	0.27 ± 0.01
RLD (ESM-2)	87.7 ± 0.3	0.32 ± 0.01
LDM (ESM3)	88.6 ± 0.3	0.28 ± 0.00
RLD (ESM3)	87.5 ± 0.1	0.32 ± 0.03

Both LDM and RLD produce sequences with high pLDDT and low pRMSD on both heavy and light chains, with all values near the natural test set. Differences across methods are small and do not track the differences seen in distributional metrics (Sec. 5.2). One pattern is worth noting. LDM scores slightly above the test set on pLDDT (e.g., 85.1 vs. 83.6 for ESM-2 heavy), while RLD matches the test set more closely (83.9 vs. 83.6). A model that collapses CDR positions to their mean produces sequences that fold cleanly precisely because they are conservative, which is consistent with the mode-collapse behaviour of LDM reported in Sec. 5.2 (CD = 0.028 on human light chains). We therefore use distributional and diversity metrics (Secs. B.1 and B.3) as our primary evaluation axes, and report structure-based metrics here for completeness rather than in the main results.

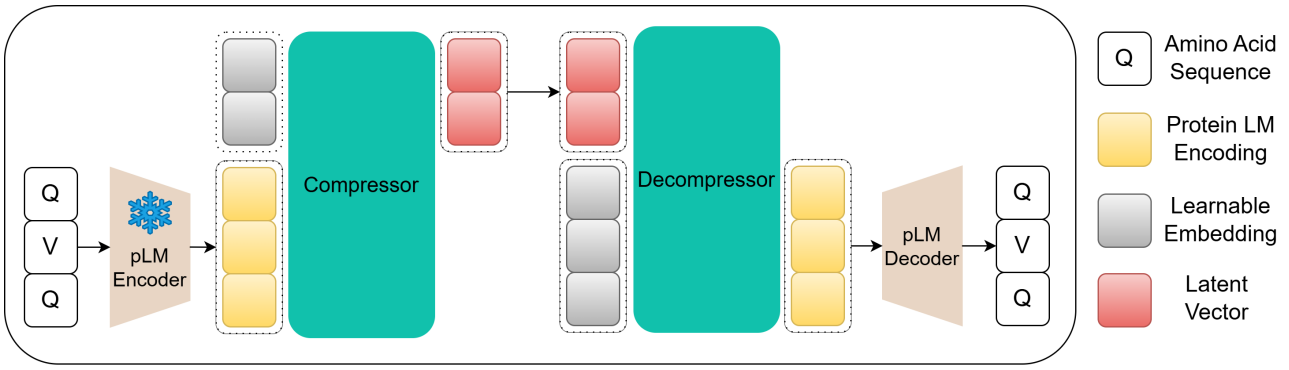


Figure 8. **Compressor–decompressor architecture.** The compressor concatenates the frozen PLM activations $h \in \mathbb{R}^{L \times d_{\text{enc}}}$ with N learnable query vectors $u \in \mathbb{R}^{N \times d}$ and processes the joint stack through a 12-layer cross-attention transformer; the latent $z \in \mathbb{R}^{N \times d}$ corresponds to the output at the query positions. The decompressor mirrors this construction: z is concatenated with a fresh set of L learnable query vectors $v \in \mathbb{R}^{L \times d}$ and processed through a 12-layer cross-attention transformer to recover $\hat{h} \in \mathbb{R}^{L \times d_{\text{enc}}}$. A linear token decoder produces amino-acid logits.

C. Architecture Details

The autoencoder maps frozen protein-language-model representations to a compact latent space and back. Given a tokenized antibody sequence $\mathbf{w} = (w_1, \ldots, w_L)$, a frozen encoder E_{PLM} produces token representations $\mathbf{h} = E_{\text{PLM}}(\mathbf{w}) \in \mathbb{R}^{L \times d_{\text{enc}}}$. We pad all sequences to a fixed maximum length $L = 154$, which covers the heavy and light chain lengths used in our experiments.

C.1. Autoencoder Architecture

The **compressor** is a 12-block cross-attention transformer. It starts from a set of randomly initialized learnable queries $\mathbf{u} \in \mathbb{R}^{N \times d}$ and treats the encoder activations $\mathbf{h}$ as the information-bearing input. In each block, queries are projected only from the latent slots, while keys and values are projected from the concatenation of the latent slots and the encoder activations:

$$\mathbf{Q} = \mathbf{u}\mathbf{W}_Q, \qquad \mathbf{K}, \mathbf{V} = [\mathbf{u}; \mathbf{h}]\mathbf{W}_{K,V}. \tag{5}$$

Stacking 12 such blocks lets each latent slot aggregate information from the full sequence while also communicating with the other latent slots through the shared key–value pool. The final latent tensor is $\mathbf{z} \in \mathbb{R}^{N \times d}$ with $N = 128$ and $d = 16$ in all experiments.

The **decompressor** is architecturally identical to the compressor, but it uses a fresh set of learnable queries $\mathbf{v} \in \mathbb{R}^{L \times d}$ that is distinct from $\mathbf{u}$. Here the information-bearing input is the latent tensor $\mathbf{z}$, and each block applies

$$\mathbf{Q} = \mathbf{v}\mathbf{W}_Q, \qquad \mathbf{K}, \mathbf{V} = [\mathbf{v}; \mathbf{z}]\mathbf{W}_{K,V}. \tag{6}$$

After 12 mirrored blocks, the decompressor outputs token-level representations $\hat{\mathbf{h}} \in \mathbb{R}^{L \times d_{\text{enc}}}$. This symmetry makes the compression and expansion pathways structurally matched while keeping the latent bottleneck explicit.

The **token decoder** is a linear projection from $\hat{\mathbf{h}}$ to amino-acid logits at each position, followed by a softmax. We train this decoder jointly with the compressor and decompressor using the autoencoder objective from Eq. (1), while the underlying PLM encoder remains frozen throughout.

C.2. Length Handling and Normalization

All sequences are padded to $L = 154$ before PLM encoding, using the EOS and PAD tokens of the underlying encoder's tokenizer. Padded positions are masked in all cross-attention so the compressor and decompressor attend only to real residues and the EOS token. The compressor concatenates the L token activations with $N = 128$ learnable query vectors and processes them through 12 cross-attention blocks, producing $z \in \mathbb{R}^{N \times d}$. The decompressor concatenates z with a fresh set of L learnable query vectors, which serve as learned positional encodings, and processes them through 12 mirrored cross-attention blocks to produce $\hat{h} \in \mathbb{R}^{L \times d_{\text{enc}}}$. A linear head maps each position to token logits. At generation time, we

decode all L positions and truncate at the first sampled EOS, so sequence length is determined implicitly by where the model places EOS rather than by an explicit length-prediction head.

Before diffusion training, we normalize each latent dimension independently by z-scoring with mean and variance estimated on a held-out subset of training sequences. This normalization makes the latent coordinates comparable in scale and improves the stability of variance-preserving diffusion.

C.3. Diffusion Denoiser

We train a Gaussian diffusion model on the normalized latent representations following standard practice (Ho et al., 2020). The forward process corrupts a clean latent $\mathbf{z}_0$ with Gaussian noise:

$$\mathbf{z}_t = \sqrt{\alpha_t}\,\mathbf{z}_0 + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{7}$$

where α_t is a monotonically decreasing noise schedule with $\alpha_0 = 1$ and $\alpha_1 \approx 0$. We train a neural denoiser $\mathbf{z}_\theta(\mathbf{z}_t, t)$ to predict the clean latent by minimizing

$$\mathcal{L}_{\text{DM}} = \mathbb{E}_{\mathbf{z}_0, t, \boldsymbol{\epsilon}} \left[\|\mathbf{z}_0 - \mathbf{z}_\theta(\mathbf{z}_t, t)\|_2^2 \right]. \tag{8}$$

The denoiser architecture follows DiMA (Meshchaninov et al., 2025b). The model consists of 12 transformer layers with 16 attention heads and a hidden dimension of 320.

The architecture incorporates several key components. Trainable positional encodings are added to the input latent representations. Time embeddings are linearly projected and summed at the input of each transformer block to condition on the diffusion timestep. Long skip connections connect shallow and deep transformer layers to facilitate learning near-identity transformations at low noise levels.

The model employs self-conditioning during training and inference. At timestep t, the previous prediction $\hat{z}_{0,s}$ from timestep $s > t$ is linearly transformed and added to each transformer block input. During training, self-conditioning is provided in 50% of cases, with a zero vector used otherwise.

D. Training Details

D.1. Pretraining

We use filtered unpaired dataset for pretraining (Sec. A.1). We train until convergence up to 1M steps, with warmup and cosine learning rate scheduler, (min lr- 1e-4, max lr- 4e-4), batch size = 512. During pretraining we also pass species class and chain type to teach class and type conditional task without additional training (condition drop probability =0.1, see Sec. F for details).

D.2. Finetuning on infilling task

We use same dataset as for pretraining. We initialize tuning from RLD checkpoint (1M pretrain steps) with N=16, D=512 and ESM3-RLD compressor space. During finetuning we take X_0 with masked CDR regions (CDR3 only or all CDRs) (C_0), encode it with the same ESM3-RLD compressor, then project C_0 into diffusion hidden size by linear projector and pass hidden C_0 into each score estimator transformer block via cross attention. We tune the model (diffusion + conditional projector) for 200k steps with batch size 512, warmup and max learning rate 2e-4. For models comparison we used randomly taken 100 paired antibodies from our test set.

D.3. Finetuning on pairing task

We use paired OAS dataset for pairing task (Sec. A.2). We start from the same checkpoint as at infilling task and keep the configuration of condition passing technically the same. The difference is that we get the C_0 from the chain pair sequence. During training we randomly choose heavy or light chain as a condition (50% propability). We tune the model for 100k steps with the same training configurations as at infilling task. For models comparison we used randomly taken 100 paired antibodies from our test set.

D.4. Finetuning on structure conditioning task

We finetune the structure-conditioned model on the antibody structure dataset of Sec. A.3. We initialize from the same RLD pretraining checkpoint (1M steps) as the infilling and pairing finetunes, with the ESM3-based RLD compressor. The conditioning input C_0 is the sequence of discrete structure tokens produced by the ESM3 structure VQ-VAE. We embed each token, project the resulting per-position vectors into the diffusion hidden size with a linear projector, and pass the projected representation into each score-estimator transformer block via cross-attention, mirroring the infilling and pairing pathways. We tune the diffusion model and the conditional projector for 100k steps with batch size 512, warmup, and a maximum learning rate of $2\mathrm{e}{-4}$.

D.5. Compute Resources

All experiments were run on NVIDIA A100 and H100 GPUs (80 GB). Autoencoder training (100k steps, batch size 512) takes approximately 3 hours on 8 A100s. Diffusion pretraining (1M steps, batch size 512) takes approximately 3 days on 8 A100s per encoder configuration. Each conditional finetune (infilling, pairing, structure-conditioning; 100k–200k steps) requires an additional 8x8 GPU-hours.

E. Additional Experiments

E.1. RLD Generalization to Different Encoder Models

We test generalization of our method on 6 encoders. We train both RLD and LDM with the same number of training steps and evaluate by generating 2048 human heavy and 2048 human light sequences. LDM model does not manage to learn the training distribution, collapsing to simple mode: very low distribution similarity to train (FD heavy 5.38-9.02 vs 0.36, light 4.39-6.22 vs 0.25) and very low diversity especially for light chain (CD light 0.028-0.066 vs 0.839) across all encoders. In contrast RLD can be successfully utilized in different encoder spaces. Full comparison values are in Tab. 7 and Fig. 9.

Table 7. **Benchmarking Encoder-based Models.** Performance comparison of Latent Diffusion Models (LDM) and RLD models across different protein language model encoders. FD values are scaled by 10^{-2}. Reported values represent the mean $\pm$ standard deviation across five independent runs of 2,048 generated sequences.

		FD ↓ (10^{-2})		PPL ↓		CD ↑	
Encoder	**Model**	Heavy	Light	Heavy	Light	Heavy	Light
Dataset		0.36 ± 0.01	0.25 ± 0.01	2.62 ± 0.01	2.75 ± 0.01	0.971 ± 0.005	0.839 ± 0.005
ESM-2	LDM	6.15 ± 1.48	4.60 ± 0.50	1.57 ± 0.01	1.59 ± 0.02	0.283 ± 0.014	0.028 ± 0.005
	RLD	0.57 ± 0.02	0.61 ± 0.03	2.30 ± 0.01	2.40 ± 0.01	0.978 ± 0.005	0.789 ± 0.007
ESM3	LDM	9.02 ± 0.80	5.71 ± 0.50	1.79 ± 0.02	1.82 ± 0.02	0.157 ± 0.015	0.037 ± 0.005
	RLD	0.49 ± 0.02	0.49 ± 0.01	2.29 ± 0.01	2.40 ± 0.01	0.957 ± 0.008	0.749 ± 0.010
IgBert	LDM	6.04 ± 0.60	4.51 ± 0.45	1.63 ± 0.02	1.60 ± 0.02	0.346 ± 0.030	0.031 ± 0.005
	RLD	0.46 ± 0.02	0.39 ± 0.01	2.39 ± 0.02	2.50 ± 0.02	0.973 ± 0.004	0.782 ± 0.023
AntiBERTy	LDM	5.38 ± 0.50	4.39 ± 0.40	1.73 ± 0.02	1.61 ± 0.02	0.679 ± 0.050	0.038 ± 0.005
	RLD	0.47 ± 0.02	0.65 ± 0.03	2.38 ± 0.02	2.32 ± 0.02	0.978 ± 0.006	0.658 ± 0.015
AbLang2	LDM	8.28 ± 0.80	6.22 ± 0.60	2.00 ± 0.02	2.14 ± 0.02	0.405 ± 0.040	0.066 ± 0.010
	RLD	0.39 ± 0.01	0.47 ± 0.21	2.42 ± 0.01	2.57 ± 0.06	0.976 ± 0.012	0.800 ± 0.013

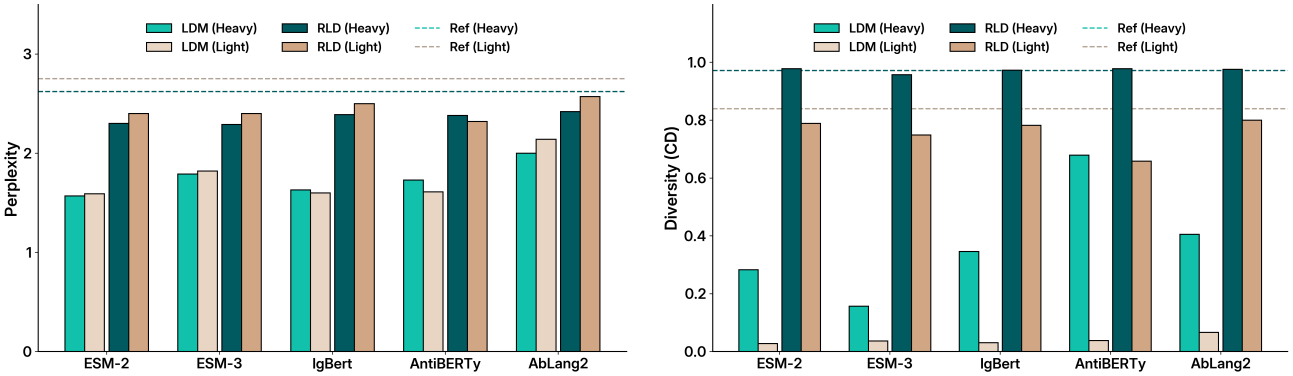


Figure 9. **Latent redistribution fixes capacity misallocation regardless of encoder architecture.**

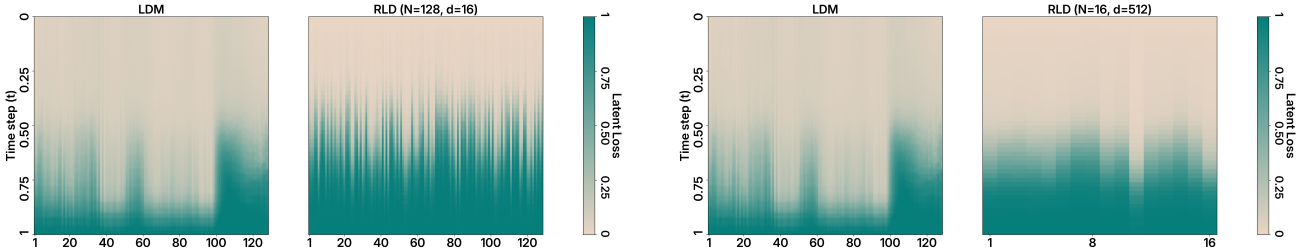
E.2. Impact of Autoencoder Dimensions

We search for the optimal autoencoder model varying hidden dimension (d) and sequence length (N). For each experiment we first train the autoencoder for 100k steps and then train diffusion for 1M steps to directly validate the diffusion usefulness of the compressor space. We use ESM-2 as the encoder for these experiments.

Table 8. **Impact of varying autoencoder hidden dimension (d) and sequence length (N) on the RLD performance.** The data is split between human heavy and light chains. Top section varies d with fixed $N = 128$; bottom section varies N with fixed $d = 512$. Reported values represent the mean and standard deviation across five independent runs of 2,048 generated sequences.

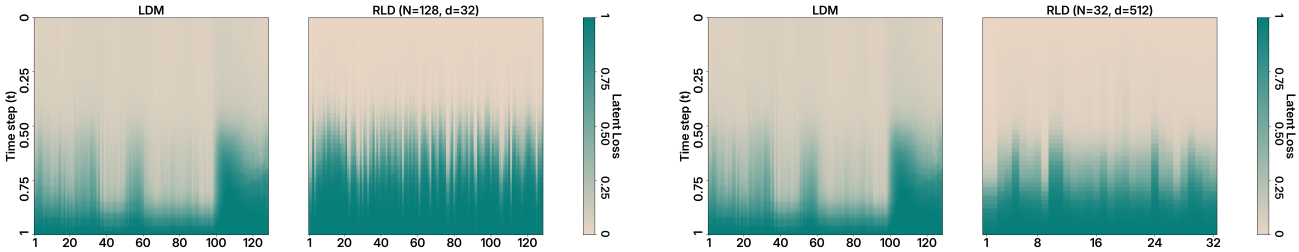
N	d	FD ↓ (10^{-2})		PPL ↓		CD ↑	
		Heavy	Light	Heavy	Light	Heavy	Light
128	16	0.38 ± 0.02	0.34 ± 0.03	2.41 ± 0.02	2.60 ± 0.01	0.970 ± 0.013	0.860 ± 0.021
128	32	0.41 ± 0.02	0.34 ± 0.01	2.39 ± 0.01	2.57 ± 0.01	0.978 ± 0.010	0.841 ± 0.009
128	64	0.39 ± 0.01	0.44 ± 0.02	2.41 ± 0.02	2.58 ± 0.01	0.977 ± 0.006	0.836 ± 0.009
128	128	0.40 ± 0.01	0.38 ± 0.06	2.42 ± 0.02	2.57 ± 0.01	0.972 ± 0.008	0.837 ± 0.006
128	256	0.39 ± 0.01	0.37 ± 0.02	2.41 ± 0.02	2.57 ± 0.01	0.971 ± 0.012	0.827 ± 0.013
128	512	0.38 ± 0.01	0.35 ± 0.01	2.44 ± 0.02	2.55 ± 0.02	0.979 ± 0.008	0.828 ± 0.007
128	1024	0.43 ± 0.01	0.36 ± 0.01	2.42 ± 0.01	2.54 ± 0.02	0.968 ± 0.008	0.808 ± 0.015
16	512	0.46 ± 0.01	0.39 ± 0.02	2.33 ± 0.01	2.50 ± 0.01	0.966 ± 0.006	0.801 ± 0.010
32	512	0.40 ± 0.01	0.38 ± 0.05	2.41 ± 0.02	2.55 ± 0.01	0.966 ± 0.006	0.822 ± 0.018
64	512	0.38 ± 0.01	0.36 ± 0.02	2.44 ± 0.01	2.55 ± 0.01	0.979 ± 0.004	0.832 ± 0.012
128	512	0.38 ± 0.01	0.35 ± 0.01	2.44 ± 0.02	2.55 ± 0.02	0.979 ± 0.008	0.828 ± 0.007
256	512	0.38 ± 0.01	0.35 ± 0.02	2.43 ± 0.01	2.59 ± 0.02	0.978 ± 0.007	0.835 ± 0.005

The quantitative metrics in Tab. 8 are accompanied by a qualitative diagnostic on ESM3. Figure 10 shows per-position diffusion loss across timesteps, the same diagnostic used in Sec. 3, for LDM and RLD across the (N, d) configurations on human heavy chains. LDM exhibits the CDR vertical-band pathology at every configuration. RLD removes the bands across the full grid. The diagnostic from Sec. 3 thus generalizes from ESM-2 to ESM3, and the redistribution effect is robust to architectural choices.



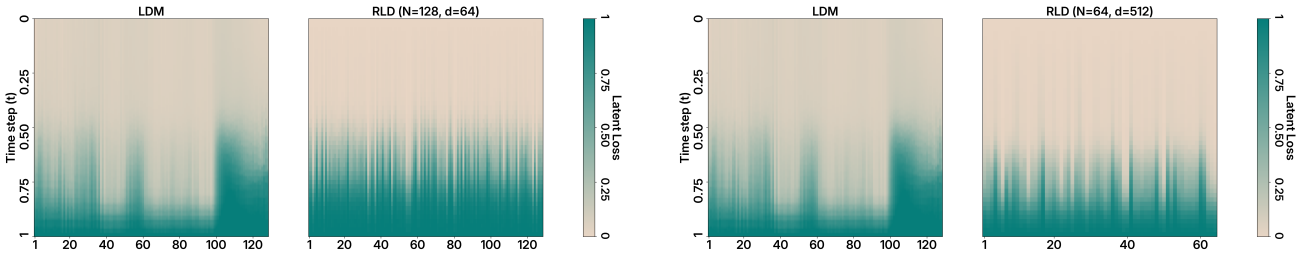
(a) $N = 128$, $d = 16$.

(b) $N = 16$, $d = 512$.



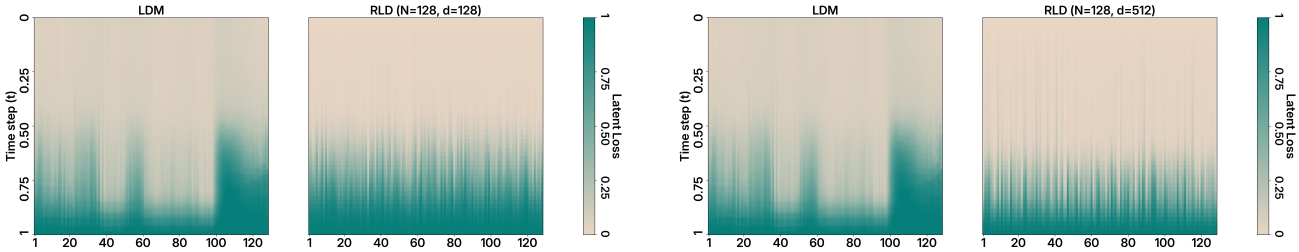
(c) $N = 128$, $d = 32$.

(d) $N = 32$, $d = 512$.



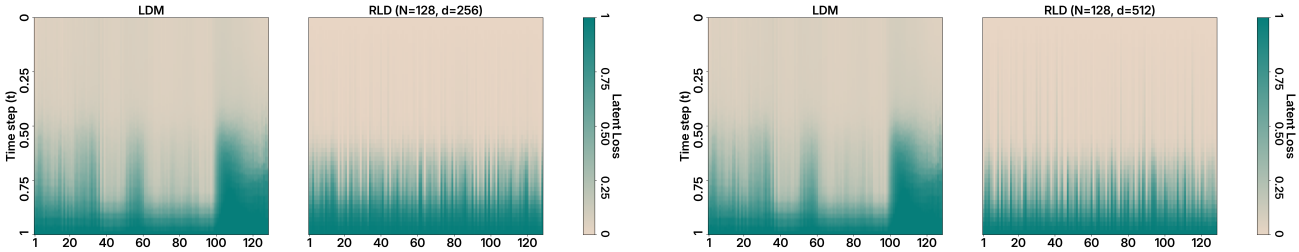
(e) $N = 128$, $d = 64$.

(f) $N = 64$, $d = 512$.



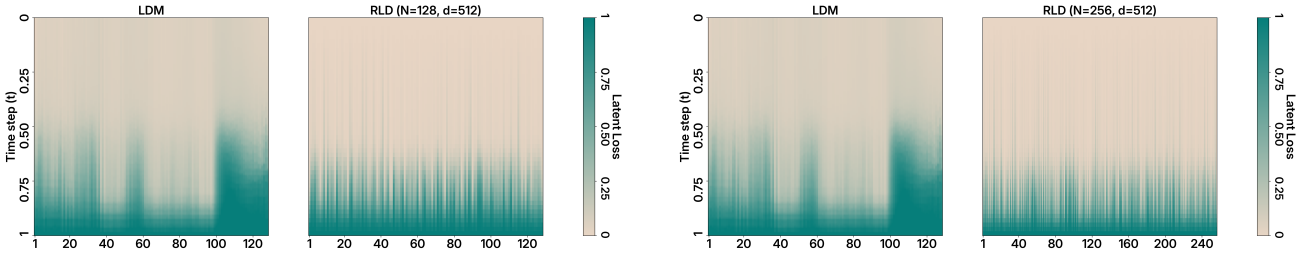
(g) $N = 128$, $d = 128$.

(h) $N = 128$, $d = 512$.



(i) $N = 128$, $d = 256$.

(j) $N = 128$, $d = 512$.



(k) $N = 128$, $d = 512$.

(l) $N = 256$, $d = 512$.

Figure 10. **ESM3 capacity misallocation across architectural choices.** Per-position diffusion loss across timesteps for ESM3-based latent diffusion models, comparing LDM (left of each pair) and RLD (right) for varying (N, d) configurations on human heavy chains. The left column varies d at $N = 128$; the right column varies N at $d = 512$. LDM consistently exhibits elevated-loss vertical bands at CDR positions. RLD removes these bands across the full grid, indicating that capacity misallocation and its resolution by redistribution are robust to the choice of (N, d).

E.3. Autoencoder Ablation

We ablate our approach by training the compressor (100k steps) and diffusion (1M steps) in multiple configurations and generate 2048 to validate the learned antibody space. All experiments are made with ESM2 as encoder.

Table 9. **Autoencoder ablation.** Evaluation of FD, Perplexity (PPL), and CD metrics across human heavy and light chains. The table also reports the diffusion loss standard deviation and Smoothness AUC. Reported values represent the mean and standard deviation across five independent runs of 2,048 generated sequences.

Model	FD ↓ (10^{-2})		PPL ↓		CD ↑		Loss std ↓ (10^{-2})		Smoothness, AUC ↓	
	Heavy	Light	Heavy	Light	Heavy	Light	Heavy	Light	Heavy	Light
Dataset	0.36 ± 0.01	0.25 ± 0.01	2.62 ± 0.01	2.75 ± 0.01	0.971 ± 0.005	0.839 ± 0.005	—	—	—	—
LDM (on ESM2)	6.15 ± 0.30	4.60 ± 0.20	1.57 ± 0.02	1.59 ± 0.02	0.283 ± 0.015	0.028 ± 0.005	8.4	8.3	—	—
RLD	1.64 ± 0.10	1.85 ± 0.36	1.97 ± 0.01	2.10 ± 0.07	0.934 ± 0.010	0.503 ± 0.016	6.8	6.7	0.38 ± 0.02	0.38 ± 0.02
RLD + aug.	1.05 ± 0.21	0.88 ± 0.01	2.15 ± 0.01	2.23 ± 0.01	0.969 ± 0.005	0.701 ± 0.011	5.4	5.3	0.30 ± 0.02	0.30 ± 0.02
RLD + aug. + drop	0.57 ± 0.02	0.61 ± 0.03	2.30 ± 0.01	2.40 ± 0.01	0.978 ± 0.005	0.789 ± 0.007	4.0	3.9	0.18 ± 0.01	0.18 ± 0.01

E.4. Latent Space Smoothness Evaluation

Sec. 4.3 evaluated latent-space smoothness on ESM-2-based RLD. Most experiments in the paper use ESM3 as the base encoder. We replicate the smoothness analysis on ESM3 to verify that smoothness transfers across encoders, and across the architectural choices of N and d. Lower AUC indicates a smoother manifold; the ESM-2 reference at the production setting ($N = 128$, $d = 16$) is 0.18 ± 0.01 on heavy chains.

Table 10. **Smoothness AUC at varying N ($d = 512$).** ESM3-based RLD. Lower is smoother. Mean and standard deviation across five independent runs.

N	Heavy	Light
16	0.21 ± 0.02	0.19 ± 0.02
32	0.17 ± 0.01	0.11 ± 0.02
64	0.19 ± 0.02	0.12 ± 0.02
128	0.15 ± 0.02	0.09 ± 0.02
256	0.16 ± 0.02	0.09 ± 0.02

Table 11. **Smoothness AUC at varying d ($N = 128$).** ESM3-based RLD. Lower is smoother. Mean and standard deviation across five independent runs.

d	Heavy	Light
16	0.25 ± 0.02	0.21 ± 0.02
32	0.27 ± 0.02	0.26 ± 0.02
64	0.19 ± 0.02	0.14 ± 0.02
256	0.15 ± 0.02	0.09 ± 0.01
512	0.15 ± 0.02	0.09 ± 0.02

ESM3-based RLD attains smoothness AUC in the same range as the ESM-2 model across all tested (N, d), indicating that the redistribution objective produces a smooth latent manifold across encoder choice and architectural settings.

E.5. Per-Position Loss Reweighting and Noise Schedules

Capacity misallocation can in principle be addressed at the level of the diffusion process rather than the representation. We evaluate two such interventions on human heavy and light chains, namely, per-position loss reweighting and an adaptive per-position noise schedule. Both modify the training of an LDM on frozen ESM-2 latents while leaving the representation unchanged. We restrict the comparison to human chains because the encoder generalization study (Sec. E.1) shows that the LDM failure mode is consistent across organisms.

Loss reweighting. Standard LDM training applies a uniform per-position MSE, allocating equal capacity to conserved and variable positions. To bias learning toward the more informative positions, we reweight the per-position loss by local sequence variability estimated from 4-gram entropy. For each IMGT-aligned position i, we collect all 4-grams $s_{i:i+4}$ across the training set and compute the Shannon entropy $H_i = -\sum_g p(g) \log p(g)$, with separate estimates for heavy and light chains. We use 4-grams as a compromise between local context and statistical reliability of the empirical $p(g)$ given the dataset size. The resulting weights are normalized so that $\sum_i w_i = L$ to preserve the overall loss scale, with the maximum-to-minimum ratio capped at 10 to prevent degenerate under-weighting of conserved positions. We found that uncapped ratios produced unstable training in preliminary runs. The weighted loss is

$$\mathcal{L} = \frac{1}{\sum_i m_i} \sum_i w_i \, m_i \, \|z_0^{(i)} - \hat{z}_0^{(i)}\|^2, \tag{9}$$

where m_i is the attention mask. The noise schedule and all other training details remain identical to the base LDM.

Adaptive noise schedule. Adapting the time-warping framework of Dieleman et al. (2022) to a per-position setting, we learn a position-specific time warping that equalizes the per-position loss profile across positions. Under a shared linear schedule, $L_i(t)$ varies substantially across positions: conserved positions have near-zero loss even at high noise levels, while CDR positions accumulate loss more rapidly. The warp table $W \in \mathbb{R}^{L \times K}$ stores per-position effective times on a grid of $K = 50$ points discretizing $[\epsilon, T]$. We chose $K = 50$ as the smallest grid that preserved the shape of the empirical loss CDFs while keeping the update step lightweight. During training, a global time t is sampled uniformly and mapped to per-position effective times $t_i^{\text{eff}} = W_i(t)$ via linear interpolation. The VP-SDE marginal parameters μ_i, σ_i are then computed from t_i^{eff} independently for each position, yielding per-position forward noising $z_t^{(i)} = \mu_i \, z_0^{(i)} + \sigma_i \, \epsilon^{(i)}$.

Every 5,000 training steps we update the warp table from accumulated per-position loss statistics. We accumulate the mean per-position loss in K time bins, compute the empirical loss CDF for each position and the mean CDF across positions, and apply CDF matching: for each position i, we find the time remapping that transforms its loss CDF to the mean CDF, computed via vectorized binary search with linear interpolation. The new warp is combined with the existing table using EMA smoothing ($\alpha = 0.9$) and made monotone via cumulative maximum. The 5,000-step interval was chosen so that loss statistics are estimated from a sufficiently large pool of batches without the warp drifting between updates. The EMA coefficient was set to the value used in Dieleman et al. (2022). Reverse sampling uses the same warp table: at each step from global time t to $t - \Delta t$, the per-position effective times are looked up and the DDPM posterior is computed with the corresponding per-position $\alpha_i(t), \alpha_i(t - \Delta t)$.

Training setup. Both methods use the same base architecture, training duration (1M steps), batch size (512), and optimizer settings as the ESM-2 LDM row in Tab. 7. We report mean and standard deviation over five runs of 2,048 generated sequences each, matching the protocol used elsewhere in the paper.

Discussion. Loss reweighting raises clustering density relative to vanilla LDM (CD H: $0.283 \to 0.566$; CD L: $0.028 \to 0.746$) but FD increases on both chains. The two effects together suggest that reweighting changes which sequences the model can produce without bringing the overall distribution closer to natural antibodies, and that getting useful behavior from per-position loss weights is a delicate calibration problem. The adaptive schedule is more effective: it reduces FD by roughly $4\times$ on heavy chains and $6\times$ on light chains relative to the LDM baseline and recovers high diversity. RLD reaches lower FD still on heavy chains (0.57 vs. 1.47, $\times 10^{-2}$) at comparable diversity, and matches the adaptive schedule on light chains. We did not perform extensive hyperparameter sweeps for either method. The adaptive schedule largely follows Dieleman et al. (2022) with the per-position extension described above, and the loss-reweighting hyperparameters (the 4-gram window, the cap, the normalization) were chosen heuristically with a small amount of exploration. More tuning could narrow the gap between RLD and the adaptive schedule. However, the structural distinction is independent of this gap. Both

Table 12. **Process-level alternatives on human heavy and light chains.** FD values are scaled by 10^{-2}. Reported values are mean $\pm$ standard deviation across five independent runs of 2,048 generated sequences. PPL is omitted; we report FD and CD as the distributional and diversity axes most relevant to the comparison.

Method	FD ↓ (10^{-2}) Heavy	FD ↓ (10^{-2}) Light	CD ↑ Heavy	CD ↑ Light
Dataset	0.36 ± 0.01	0.25 ± 0.01	0.971 ± 0.005	0.839 ± 0.005
LDM (ESM-2)	6.15 ± 1.48	4.60 ± 0.50	0.283 ± 0.014	0.028 ± 0.005
+ loss reweighting	12.85 ± 1.20	11.56 ± 0.85	0.566 ± 0.020	0.746 ± 0.015
+ adaptive schedule	1.47 ± 0.06	0.75 ± 0.03	0.986 ± 0.004	0.800 ± 0.006
RLD (ESM-2)	0.57 ± 0.02	0.61 ± 0.03	0.978 ± 0.005	0.789 ± 0.007

process-level methods rely on per-position statistics defined on a fixed positional coordinate system (here, IMGT-aligned positions), whereas RLD learns redistribution end-to-end without requiring such an alignment.

F. Organism-Conditioned Generation

We condition the diffusion model on organism class and chain type during training. Both conditioning signals are embedded into the same hidden dimension as X_t and added to the latent representation between each score estimator transformer block. During training, we randomly replace the organism class and chain type tokens with special `<all_class>` and `<all_type>` tokens with independent 10% probabilities. This allows the model to optionally generate without explicit organism or chain type specification.

Table 13. **Human Antibody Generation.** Evaluation of FD, Perplexity (PPL), and CD metrics for Human Heavy and Light chains. Values represent mean and standard deviation.

Chain	Metric	Dataset	DiMA	EAGLE	IgLM	MD (DiffAb)	RLD	AbBFN2	IgCraft	p-IgGen
Human Heavy	FD ↓	0.0034 ± 0.0001	0.0615 ± 0.0148	0.0759 ± 0.0010	0.0987 ± 0.0008	0.0318 ± 0.0008	0.0038 ± 0.0002	0.0174 ± 0.0006	0.0198 ± 0.0005	0.0218 ± 0.0006
	PPL ↓	2.58 ± 0.03	1.57 ± 0.01	1.63 ± 0.01	2.70 ± 0.01	3.28 ± 0.01	2.41 ± 0.02	2.09 ± 0.01	2.03 ± 0.01	1.96 ± 0.01
	CD ↑	0.972 ± 0.005	0.283 ± 0.014	0.204 ± 0.005	0.986 ± 0.001	0.997 ± 0.001	0.970 ± 0.013	0.959 ± 0.005	0.942 ± 0.005	0.931 ± 0.001
Human Light	FD ↓	0.0031 ± 0.0004	0.0460 ± 0.0050	0.0500 ± 0.0050	0.0056 ± 0.0004	0.0195 ± 0.0001	0.0034 ± 0.0003	0.0234 ± 0.0007	0.0219 ± 0.0007	0.0252 ± 0.0007
	PPL ↓	2.77 ± 0.02	1.59 ± 0.01	1.59 ± 0.01	2.46 ± 0.01	3.14 ± 0.01	2.60 ± 0.01	1.94 ± 0.01	1.91 ± 0.01	1.93 ± 0.01
	CD ↑	0.901 ± 0.009	0.028 ± 0.005	0.027 ± 0.005	0.823 ± 0.006	0.901 ± 0.002	0.860 ± 0.021	0.368 ± 0.004	0.317 ± 0.003	0.298 ± 0.002

Table 14. **Cross-Species Antibody Generation.** Evaluation of FD, PPL, and CD metrics across Mouse, Rhesus, Rabbit, Rat, and Camel chains.

Chain	Metric	Dataset	DiMA	EAGLE	IgLM	MD (DiffAb)	RLD
Mouse Heavy	FD ↓	0.0026 ± 0.0001	0.0180 ± 0.0020	0.0256 ± 0.0020	0.0095 ± 0.0005	0.0195 ± 0.0009	0.0025 ± 0.0002
	PPL ↓	2.26 ± 0.01	1.74 ± 0.01	1.69 ± 0.01	2.13 ± 0.01	2.46 ± 0.01	2.16 ± 0.01
	CD ↑	0.917 ± 0.011	0.222 ± 0.010	0.137 ± 0.010	0.884 ± 0.001	0.954 ± 0.005	0.885 ± 0.004
Mouse Light	FD ↓	0.0027 ± 0.0001	0.0357 ± 0.0030	0.0445 ± 0.0040	0.0322 ± 0.0030	0.0382 ± 0.0034	0.0067 ± 0.0005
	PPL ↓	2.27 ± 0.01	1.92 ± 0.02	2.02 ± 0.02	2.56 ± 0.01	2.92 ± 0.02	2.76 ± 0.02
	CD ↑	0.918 ± 0.008	0.074 ± 0.005	0.063 ± 0.005	0.544 ± 0.012	0.548 ± 0.010	0.568 ± 0.018
Rhesus Heavy	FD ↓	0.0033 ± 0.0001	0.0956 ± 0.0050	0.0969 ± 0.0050	0.0129 ± 0.0043	0.2208 ± 0.0030	0.0042 ± 0.0001
	PPL ↓	3.33 ± 0.02	2.27 ± 0.02	2.48 ± 0.02	2.95 ± 0.01	3.70 ± 0.03	3.28 ± 0.01
	CD ↑	0.905 ± 0.005	0.116 ± 0.010	0.103 ± 0.010	0.976 ± 0.001	0.999 ± 0.001	0.995 ± 0.002
Rhesus Light	FD ↓	0.0034 ± 0.0002	0.0657 ± 0.0050	0.0632 ± 0.0050	0.0105 ± 0.0032	0.0995 ± 0.0036	0.0043 ± 0.0001
	PPL ↓	3.32 ± 0.02	2.09 ± 0.02	2.01 ± 0.02	2.70 ± 0.01	3.54 ± 0.03	2.97 ± 0.01
	CD ↑	0.902 ± 0.009	0.104 ± 0.010	0.075 ± 0.005	0.792 ± 0.006	0.994 ± 0.001	0.954 ± 0.003
Rabbit Heavy	FD ↓	0.0022 ± 0.0006	0.0325 ± 0.0030	0.0367 ± 0.0030	0.0030 ± 0.0003	0.0265 ± 0.0006	0.0025 ± 0.0001
	PPL ↓	6.56 ± 0.02	6.14 ± 0.05	6.07 ± 0.05	6.47 ± 0.01	6.92 ± 0.05	6.46 ± 0.01
	CD ↑	0.936 ± 0.007	0.281 ± 0.010	0.233 ± 0.010	0.882 ± 0.002	1.000 ± 0.001	0.932 ± 0.002
Rat Heavy	FD ↓	0.0018 ± 0.0001	0.0429 ± 0.0040	0.0539 ± 0.0050	0.0030 ± 0.0001	0.0097 ± 0.0002	0.0021 ± 0.0001
	PPL ↓	2.34 ± 0.01	1.70 ± 0.01	1.66 ± 0.01	2.17 ± 0.01	2.63 ± 0.02	2.26 ± 0.01
	CD ↑	0.966 ± 0.004	0.058 ± 0.005	0.033 ± 0.005	0.957 ± 0.001	1.000 ± 0.001	0.983 ± 0.001
Camel Heavy	FD ↓	0.0037 ± 0.0001	0.0926 ± 0.0090	0.1141 ± 0.0100	0.0098 ± 0.0007	0.0348 ± 0.0007	0.0094 ± 0.0013
	PPL ↓	5.53 ± 0.03	4.22 ± 0.04	4.32 ± 0.04	4.95 ± 0.01	6.22 ± 0.05	5.56 ± 0.05
	CD ↑	0.979 ± 0.002	0.992 ± 0.001	0.950 ± 0.005	1.000 ± 0.001	1.000 ± 0.001	1.000 ± 0.001

To rule out memorization, we compute the maximum sequence identity between each generated sequence and its nearest neighbor in the training set. The training set is based on the IgLM dataset, clustered at the 95% identity threshold. Lower values indicate sequences further from any training example, and we use the held-out test set as a reference for the natural level of nearest-neighbor identity to expect.

Across organisms and chain types, RLD's nearest-neighbor identity is at or below the level of the held-out test set, indicating that the model produces sequences with the same level of novelty relative to training as natural test sequences. The edit distances reported for paired generation in Tab. 3 (VH: 0.43, VL: 0.40) are consistent with this picture.

Table 15. **Nearest-neighbor identity to the training set.** Maximum sequence identity between each generated sequence and its nearest neighbor in the training set, averaged across 2,048 generated sequences and reported as mean ± standard deviation across five runs. Lower values indicate greater novelty. Light-chain columns are absent for Rabbit, Rat, and Camel because OAS does not contain light-chain data for these organisms.

	Human		Mouse		Rabbit	Rat	Rhesus		Camel
Model	Heavy	Light	Heavy	Light	Heavy	Heavy	Heavy	Light	Heavy
Test set	0.94 ± 0.001	0.95 ± 0.001	0.96 ± 0.001	0.95 ± 0.001	0.93 ± 0.002	0.95 ± 0.001	0.91 ± 0.000	0.93 ± 0.000	0.85 ± 0.002
LDM (ESM3)	0.92 ± 0.000	0.94 ± 0.001	0.97 ± 0.001	0.96 ± 0.001	0.92 ± 0.001	0.95 ± 0.001	0.91 ± 0.001	0.93 ± 0.001	0.83 ± 0.002
RLD (ESM3)	0.90 ± 0.001	0.93 ± 0.000	0.96 ± 0.000	0.94 ± 0.001	0.90 ± 0.001	0.91 ± 0.000	0.85 ± 0.000	0.90 ± 0.001	0.75 ± 0.001
IgLM	0.92 ± 0.001	0.93 ± 0.001	0.96 ± 0.001	0.94 ± 0.001	0.91 ± 0.001	0.93 ± 0.001	0.86 ± 0.001	0.92 ± 0.001	0.78 ± 0.002

G. Sequence Infilling

CDR infilling tests whether a model has learned the joint distribution over framework and hypervariable regions. Unlike unconditional generation, infilling requires the model to condition on surrounding context and regenerate masked regions consistent with that context. This capability is essential for practical antibody design workflows, where researchers often wish to optimize specific CDRs while preserving framework compatibility.

We evaluate three infilling scenarios:

- **Variable-length (CDR3 only):** We mask the CDR3 region without specifying its length, and require the model to regenerate both the content and the length of the missing span (Sec. G.1).
- **Single-span fixed-length (CDR3 only):** We mask the CDR3 region with its length given, and require the model to regenerate the content (Sec. G.2). CDR3 is the most variable region and the primary determinant of antigen specificity.
- **Multi-span fixed-length (all CDRs):** We mask all three CDR regions simultaneously with their lengths given, and require the model to regenerate them jointly (Sec. G.3). This tests whether the model can coordinate multiple hypervariable regions while respecting framework constraints.

For each masked sequence we generate $K = 100$ samples. The metrics differ between the variable-length and fixed-length settings and are defined in their respective subsections below.

G.1. Variable-Length CDR3 Infilling

In the variable-length setting the model receives the framework regions and CDR1/CDR2 as context, and must produce both the content and the length of the missing CDR3. This matches typical antibody design workflows more closely than the fixed-length variants, where the masked span has a known length.

We report three metrics:

- **Edit distance to reference.** Mean normalized Levenshtein distance between each generated CDR3 and the ground-truth CDR3, averaged across the test set. Lower is better.
- **Sequence divergence between samples.** For each masked sequence we compute the mean pairwise normalized Levenshtein distance over all sample pairs, then average across the test set. Higher values indicate more varied predictions for the same input.
- **ProGen2 perplexity.** Computed on the full completed sequence, as defined in Sec. B.2.

We compare against AbBFN2 and IgLM, the two baselines that natively predict variable-length CDR3 regions. AbLang, AbLang2, and ESM3 are excluded from this setting because the variable-length protocol requires sampling without a fixed mask length.

Table 16 reports results on human heavy and light chains, including a Random baseline that draws each CDR3 position uniformly from the 20 amino acids. RLD attains the lowest edit distance on both chains and matches the best perplexity on light chains, while sample divergence is comparable to the other models. All three models are well separated from the Random floor on every metric.

Table 16. **Variable-length CDR3 infilling, full results.** Edit distance to reference, sequence divergence between samples, and ProGen2 perplexity on human heavy (H) and light (L) chains.

	Edit dist. ↓		Seq. div. ↑		PPL ↓	
Model	H	L	H	L	H	L
Eval set	0.00	0.00	0.00	0.00	2.59	2.89
Random	0.96	0.99	0.95	0.95	3.21	3.71
RLD (Ours)	**0.74**	**0.39**	0.83	0.60	2.59	**2.87**
AbBFN2	0.79	0.40	**0.84**	0.62	2.59	**2.87**
IgLM	0.75	0.42	0.82	**0.67**	**2.55**	2.89

G.2. Single-Span Fixed-Length CDR3 Infilling

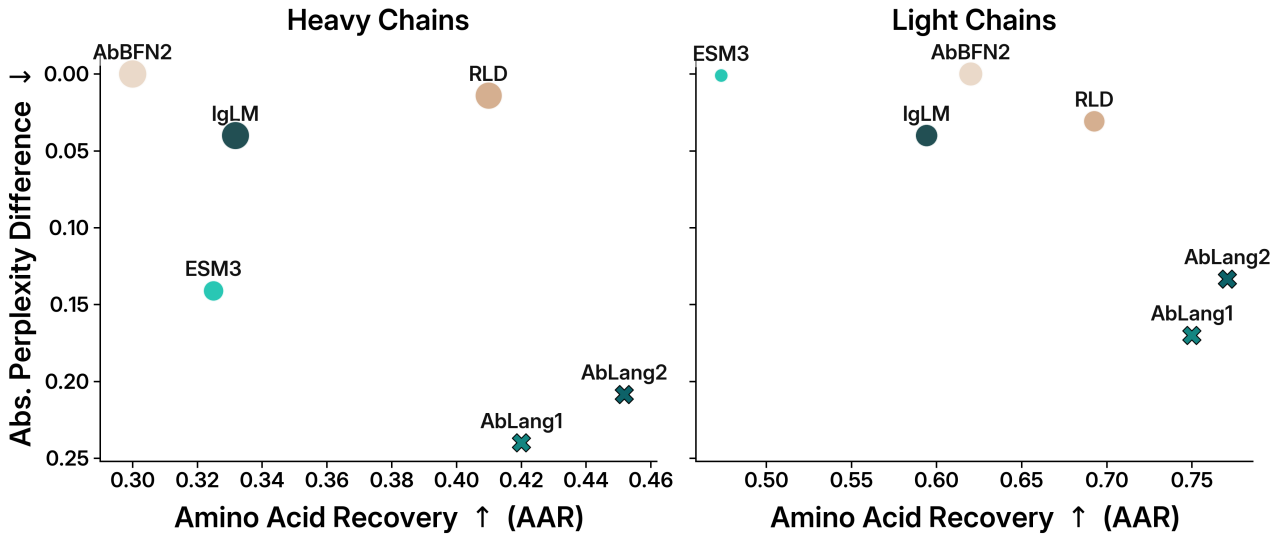


Figure 11. **CDR3 infilling performance across heavy and light chains.** Models regenerate masked CDR3 regions conditioned on framework context. This task presents a Pareto trade-off between **fidelity** (Amino Acid Recovery, higher is better) and **generation quality** (Absolute Perplexity Difference from natural sequences, lower is better). **Marker size**: diversity score (larger indicates more diverse predictions; AbLang models have zero diversity due to argmax sampling). RLD achieves strong performance on both objectives.

In the single-span fixed-length setting the model regenerates a CDR3 region of known length, given the framework and CDR1/CDR2 context. We report three metrics that we also use for the multi-span fixed-length setting:

- **Amino Acid Recovery (AAR):** The fraction of positions where the generated sequence exactly matches the ground truth. This measures exact reconstruction fidelity.
- **Absolute Perplexity Difference:** The absolute difference between the perplexity of generated sequences and natural sequences from the test set, as measured by a reference language model. This quantifies how natural the generated sequences appear.
- **Diversity Score:** Measures variability across the K samples. See Sec. B.4 for computational details.

These metrics capture different aspects of infilling quality. AAR measures exact match to the ground truth, though we note that multiple CDR sequences could be functionally valid with the same framework. Perplexity measures statistical naturalness. Diversity measures exploration versus memorization. Strong infilling performance requires balancing all three objectives, as visualized in the Pareto trade-off framework of Fig. 11.

Table 17 shows detailed CDR3 infilling results. RLD achieves 42–44% AAR on heavy chains and 68–71% AAR on light chains, with perplexity differences below 0.03 and diversity scores above 0.85. This combination indicates RLD balances accurate reconstruction with natural sequence generation and diverse sampling.

The performance gap between heavy and light chains is expected. Light chain CDR3 regions are significantly shorter (9–11 residues versus 12–15 for heavy chains) and exhibit lower sequence diversity, making them intrinsically easier to reconstruct from context.

Baseline models show distinct patterns. AbLang and AbLang2 achieve the highest AAR scores (70–75% on heavy chains) but exhibit zero diversity due to argmax decoding and elevated perplexity differences (0.13–0.25). This suggests strong memorization of training sequences but limited generalization.

Generative baselines (AbBFN2, ESM3) achieve lower AAR (30–40%) but maintain perplexity near natural sequences. The general-purpose model ESM3 shows particularly low recovery on heavy chains, likely reflecting limited antibody-specific training data. These models prioritize generating statistically plausible sequences over exact reconstruction.

Table 17. **CDR3 infilling task performance comparison. There are gaps in Diversity cells for Ablang and Ablang2, because these models use argmax for sampling, so they do not have variability.**

	AAR ↑		Diversity ↑		Exact match ↓		|ΔPPL| ↓	
Model	Heavy	Light	Heavy	Light	Heavy	Light	Heavy	Light
Dataset	1.00	1.00	0.00	0.00	1.00	1.00	0.000	0.000
Random	0.050 ± 0.001	0.050 ± 0.001	0.95 ± 0.08	0.95 ± 0.08	0.00	0.00	0.623 ± 0.003	0.824 ± 0.003
RLD (ours)	0.408 ± 0.006	0.689 ± 0.004	0.77 ± 0.06	0.52 ± 0.06	0.02	0.05	0.010 ± 0.000	0.028 ± 0.001
AbLang1	0.419 ± 0.000	0.752 ± 0.000	-	-	0.00	0.15	0.240 ± 0.000	0.162 ± 0.000
AbLang2	0.452 ± 0.000	0.771 ± 0.000	-	-	0.01	0.19	0.207 ± 0.000	0.133 ± 0.000
AbBFN2	0.305 ± 0.003	0.622 ± 0.005	0.82 ± 0.05	0.62 ± 0.08	0.00	0.02	0.001 ± 0.000	0.000 ± 0.001
IgLM	0.332 ± 0.004	0.594 ± 0.005	0.81 ± 0.05	0.55 ± 0.08	0.00	0.00	0.039 ± 0.001	0.035 ± 0.001
ESM3	0.325 ± 0.007	0.474 ± 0.007	0.47 ± 0.05	0.26 ± 0.10	0.00	0.00	0.139 ± 0.001	0.001 ± 0.003

G.3. Multi-Span Fixed-Length All-CDR Infilling

Table 18 presents results for infilling all three CDRs simultaneously, using the same metrics as the single-span fixed-length setting. RLD maintains competitive performance across both single and multiple masked regions, with AAR and perplexity metrics consistent with the single-span setting. This demonstrates that the model has learned coordinated representations of framework–CDR relationships rather than treating each region independently.

The consistency across infilling scenarios validates that RLD can handle flexible conditioning patterns, a key advantage of non-autoregressive diffusion models over autoregressive alternatives.

Table 18. **All CDRs infilling (fixed lengths) task performance comparison. There are gaps in Diversity cells for Ablang and Ablang2, because these models use argmax for sampling, so they do not have variability.**

	AAR ↑		Diversity ↑		Exact match ↓		|ΔPPL| ↓	
Model	Heavy	Light	Heavy	Light	Heavy	Light	Heavy	Light
Dataset	1.00	1.00	0.00	0.00	1.00	1.00	0.000	0.000
Random	0.045 ± 0.002	0.045 ± 0.003	0.90 ± 0.005	0.91 ± 0.007	0.00	0.00	3.191 ± 0.005	5.609 ± 0.005
RLD (ours)	0.609 ± 0.002	0.713 ± 0.002	0.68 ± 0.005	0.59 ± 0.005	0.00	0.00	0.028 ± 0.000	2.860 ± 0.003
AbBFN2	0.522 ± 0.003	0.637 ± 0.002	0.73 ± 0.003	0.67 ± 0.006	0.00	0.00	0.037 ± 0.001	2.913 ± 0.002
AbLang1	0.636 ± 0.000	0.764 ± 0.000	-	-	0.04	0.00	0.433 ± 0.000	0.371 ± 0.000
AbLang2	0.644 ± 0.000	0.778 ± 0.000	-	-	0.00	0.06	0.423 ± 0.000	0.357 ± 0.000
ESM3	0.482 ± 0.003	0.514 ± 0.003	0.51 ± 0.006	0.35 ± 0.008	0.00	0.00	0.101 ± 0.001	0.031 ± 0.000

G.4. Implications

The strong infilling performance demonstrates that RLD has learned a bidirectional understanding of antibody sequences. Unlike autoregressive models that condition only on past tokens, RLD can condition on arbitrary surrounding context to generate masked regions. This flexibility enables practical design workflows such as CDR optimization given fixed frameworks or complementary CDR generation given partial sequence information.

H. Structure-Conditioned Generation

H.1. Test-Set Construction

We evaluate on a subset of the AntiFold/AbMPNN test split. Our evaluation protocol refolds each generated sequence with IgFold (Ruffolo et al., 2023) and measures the resulting structure against the reference PDB. For test structures where IgFold cannot reliably refold the *native* sequence, the resulting RMSD measurements reflect IgFold prediction error rather than the model under evaluation. We therefore retain only those test structures for which the IgFold-refolded native sequence is within 1 Å CDR3 RMSD of the reference, ensuring that at least one sequence is known to satisfy the structural constraint. The same filtering strategy was used in Strashnov et al. (2026) (with ESMFold) to construct a solvable motif-scaffolding benchmark. From the retained pool we randomly sample 512 heavy and 512 light chains. The same set is used for all baselines, and the filter is applied to native sequences rather than to any model's generations, so no model sees a more favourable subset of the test split.

For each generated sequence, RMSD is computed as follows. We fold the generated sequence with IgFold to obtain Cα coordinates and extract the corresponding Cα from the native PDB. Both sequences are IMGT-numbered to identify shared positions. Global RMSD is computed by Kabsch alignment over all shared Cα. CDR3 RMSD is computed by Kabsch alignment over framework Cα only, with the resulting transformation applied to the CDR3 region (IMGT positions 105–117) and RMSD measured over those positions.

H.2. RMSD Results

The full RMSD results are reported in Tab. 4 of the main text. RLD and AntiFold lead on both metrics, with RLD best on CDR3 RMSD for both heavy and light chains and AntiFold best on Global RMSD. ESM3 follows as the next-strongest baseline, ahead of the antibody-specialized AbMPNN on Global RMSD for both chains. The general-protein inverse folding methods (ESM-IF1, ProteinMPNN) sit furthest from the reference, with ESM-IF1 showing the largest Global RMSD on light chains (5.07 Å) and ProteinMPNN second-largest (3.86 Å).

H.3. Per-Region Amino-Acid Recovery

Tab. 19 reports per-region AAR after IMGT numbering of generated and reference sequences. RLD achieves the highest AAR in 11 of 12 region–chain combinations. The single exception is heavy CDR3, where AntiFold leads (0.599 vs. RLD's 0.545). On light CDR3, RLD recovers 0.908 against AntiFold's 0.843. Framework regions (FR1–FR3) approach near-perfect recovery for RLD on both chains.

Table 19. **Per-region amino-acid recovery for structure-conditioned generation.** AAR per IMGT region for heavy (H) and light (L) chains on the filtered AntiFold/AbMPNN test split (512 H + 512 L). Best per column in bold.

	CDR1		CDR2		CDR3		FR1		FR2		FR3	
Model	H	L	H	L	H	L	H	L	H	L	H	L
RLD (Ours)	**0.956**	**0.932**	**0.943**	**0.958**	0.545	**0.908**	**0.988**	**0.976**	**0.978**	**0.981**	**0.977**	**0.984**
AntiFold	0.893	0.861	0.887	0.872	**0.599**	0.843	0.896	0.912	0.917	0.942	0.940	0.951
AbMPNN	0.778	0.638	0.698	0.689	0.506	0.627	0.884	0.813	0.828	0.808	0.845	0.842
ESM3	0.748	0.560	0.613	0.563	0.482	0.573	0.870	0.772	0.788	0.794	0.795	0.794
ESM-IF1	0.457	0.291	0.433	0.407	0.210	0.357	0.488	0.470	0.482	0.510	0.513	0.548
ProteinMPNN	0.523	0.300	0.478	0.568	0.145	0.450	0.566	0.549	0.527	0.584	0.563	0.659

H.4. Broader Impacts

Generative models for antibody sequences can accelerate the discovery of therapeutic antibodies for infectious disease, oncology, and autoimmune indications, and can reduce the experimental burden of repertoire exploration. We see this as the primary intended impact of our work. Direct dual-use risk is limited: our model is trained on natural antibody repertoires from OAS and generates sequences conditioned on organism, chain type, framework context, or backbone structure. It does not condition on antigen identity and does not perform affinity maturation, so it cannot be straightforwardly used to design antibodies targeting a chosen pathogen or human protein without substantial additional pipelines (antigen-conditioned design, in vitro screening, developability assessment). Indirect risks include the general risk that improved generative tooling for biomolecules lowers the barrier to misuse over time; we believe that releasing models trained only on natural repertoires, without antigen-conditioning, is consistent with current community norms for responsible release of antibody-design tools.