Antibody Generation via Redistributed Latent Diffusion

Andrey Shevtsov[*]
AXXX
Moscow

Viacheslav Meshchaninov[*]
Constructor University
Bremen

Pavel Strashnnov[*]
AXXX
Moscow

Dmitry Vetrov
Constructor University
Bremen

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

[*]Equal contribution. Correspondence: strashnov@axxx.tech

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

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 latent spaces with explicit capacity regularization, to "smooth" the entropy landscape. Its cross-attention compressor aggregates information across input positions into learned latent slots, while regularization encourages distributed representations. Together, these mechanisms reduce the imbalance in denoising difficulty 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). *(d)* We publicly release source code and datasets.

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].

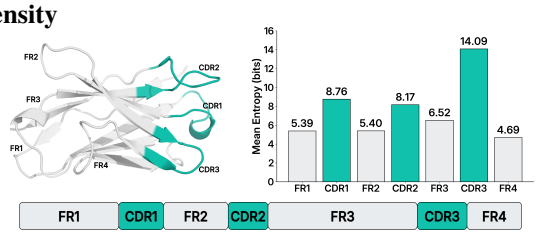


Figure 1: **Information Density Imbalance in Antibody Sequences.** *(Left)* Variable-domain structure with framework regions (FR1–FR4) and complementarity-determining regions (CDR1–CDR3). *(Bottom)* Sequence organization with segment lengths proportional to mean lengths in natural repertoires. *(Right)*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

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 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 sequences 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.

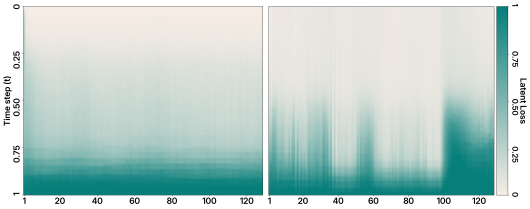


Figure 2: **Per-position diffusion loss across timesteps for ESM-2-based latent diffusion models.** *(Left)* Training on general proteins yields uniform loss across positions. *(Right)* Training on antibodies reveals capacity misallocation, with elevated loss (teal bands) concentrated in CDR regions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

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 four times the spatial variation in learning difficulty. Second,

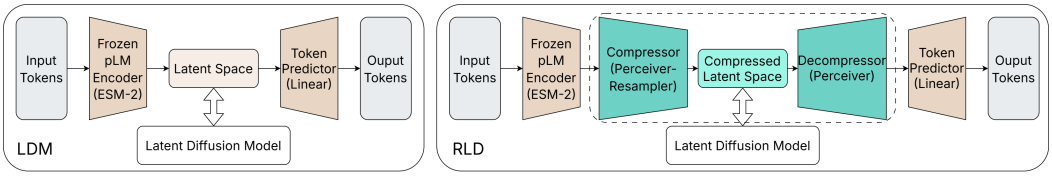


Figure 3: **Latent Diffusion vs. Redistributed Latent Diffusion.** *(Left)* LDM diffuses directly on frozen pLM hidden states. *(Right)* RLD inserts a learned compressor and decompressor, so diffusion operates on a fixed-length latent space that redistributes information across learned slots through cross-attention.

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.6. 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 architectural mechanism is cross-attention: each learned latent query aggregates information from the full input sequence, without an enforced correspondence between latent slots and input positions. This allows information to be redistributed across a fixed set of latent slots without requiring $N < L$. 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. Throughout this work, ESM-2 refers to the 150M-parameter model with $d_{\text{enc}} = 640$, except when explicitly used as an alternative evaluation encoder. 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.

Our default configuration uses $N = 128$ and $d = 16$. Cross-attention allows each latent slot to aggregate information across framework and CDR positions, while the small feature dimension keeps the latent space compact. Redistribution does not require a sequence-axis bottleneck: our ablations also show improvements with an overcomplete representation. Dimension and regularization ablations appear in Sec. E.2 and Sec. E.3.

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.

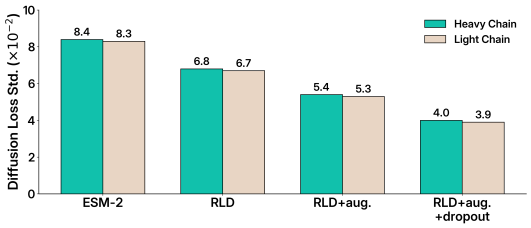


Figure 4: **Training components improve capacity uniformity.** Standard deviation of per-dimension diffusion loss across training configurations. Lower variance indicates more uniform capacity allocation. Results for human heavy chains; full metrics in Sec. E.3.

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 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. 10 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. 10 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.

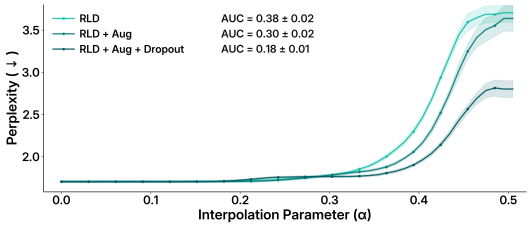


Figure 5: **Training components improve latent-space smoothness.** ProGen2 perplexity along linear interpolations between encoded sequence pairs, for three autoencoder configurations. Lower curves and lower AUC indicate smoother manifolds. Results for human heavy chains; full metrics in Sec. E.3.

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 the tested configurations (Sec. E.2), each of which retains compression along at least one axis. A separate ablation shows that improvements also persist without dimensional compression (Sec. E.3).

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. 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.6.

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. 15.

	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

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

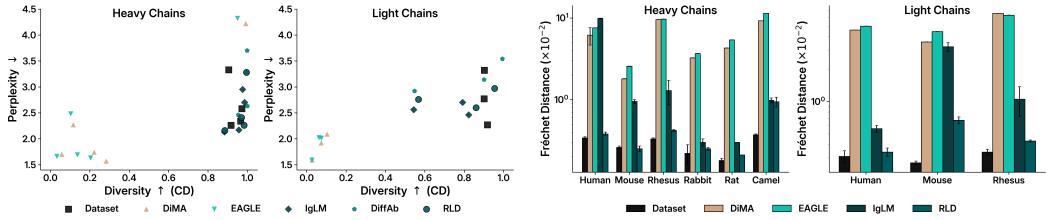


Figure 6: **Organism-conditioned generation.** *(Left)* Perplexity vs. Diversity (CD), one point per organism. Standard LDMs (DiMA, EAGLE) cluster in the low-CD, low-PPL region. RLD clusters with natural sequences. *(Right)* Fréchet distance across organisms and chain types. Standard LDMs show up to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

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. Additionally, Sec. E.7 shows that the FD results hold across five alternative evaluation embedding spaces. We assess computational developability-related properties using structure-derived metrics (Appendix E.8).

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 and 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 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. 8 reports corresponding shifts in PPL and CD).

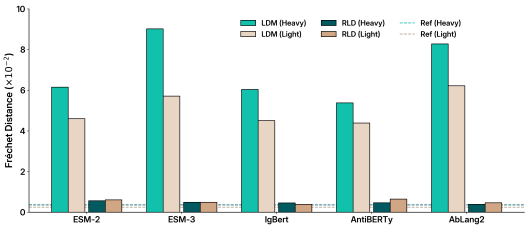


Figure 7: **Latent redistribution fixes capacity misallocation regardless of encoder architecture.** Fréchet Distance (FD, lower is better) for diffusion models trained on five encoders, comparing base latents (lighter bars) against RLD-redistributed latents (darker bar). Dashed lines indicate dataset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

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. 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.

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

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-light compatibility from learned features of natural pairings (Sec. B.7). $|\Delta\text{PPL}|$ 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 and Wei, 2023], plus two reference conditions: *TP* (True Pairs from the test set) and *FP* (False Pairs created by randomly matching training sequences).

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 ↑		$\lvert\Delta\text{PPL}\rvert$ ↓		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

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 ($|\Delta\text{PPL}|$: 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.

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

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, particularly the general-protein methods ESM-IF1 and ProteinMPNN on heavy 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. 24). 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. Full results are provided in Sec. H.

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.

References

B. Abanades, W. K. Wong, F. Boyles, G. Georges, A. Bujotzek, and C. M. Deane. 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. 15

J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022. 4

T. Atkinson, T. D. Barrett, S. Cameron, B. Guloglu, M. Greenig, L. Robinson, A. Graves, L. Copoiu, and A. Laterre. Protein sequence modelling with bayesian flow networks. *Nature Communications*, 16, 2024. 2

A. M.-J. S. M. A. S. E. B. M. O. M. A. A. Bowle, A. Bateman, M. J. Martin, S. Orchard, M. Magrane, A. Y. Adesina, S. Ahmad, E. H. Bowler-Barnett, H. Bye-A-Jee, D. Carpentier, P. Denny, J. Fan, P. Garmiri, L. J. da Costa Gonzales, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, V. Joshi, D. Jyothi, S. Kandasaamy, A. Lock, A. Luciani, J. Luo, Y. Lussi, J. S. M. Marin, P. Raposo, D. Rice, R. Santos, E. Speretta, J. L. Stephenson, P. Totoo, N. Tyagi, N. Urakova, P. Vasudev, K. Warner, S. Wijerathne, C. W.-H. Yu, R. Zaru, A. J. Bridge, L. Aimo, G. Argoud-Puy, A. H. Auchincloss, K. B. Axelsen, P. Bansal, D. Baratin, T. M. B. Neto, M.-C. Blatter, J. T. Bolleman, E. Boutet, L. Breuza, B. C. Gil, C. Casals-Casas, K. C. Echioukh, E. Coudert, B. Cuche, E. de Castro, A. Estreicher, M. L. Famiglietti, M. Feuermann, E. Gasteiger, P. Gaudet, S. Gehant, V. B. Gerritsen, A. Gos, N. Gruaz, C. Hulo, N. Hyka-Nouspikel, F. Jungo, A. Kerhornou, P. L. Mercier, D. Lieberherr, P. Masson, A. Morgat, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, S. Poux, M. Pozzato, M. Pruess, N. Redaschi, C. Rivoire, C. J. A. Sigrist, K. Sonesson, S. Sundaram, A. Sveshnikova, C. H. Wu, C. N. Arighi, C. Chen, Y. Chen, H. Huang, K. Laiho, M. Lehvaslaiho, P. B. McGarvey, D. A. Natale, K. Ross, C. R. Vinayaka, Y. Wang, and J. Zhang. Uniprot: the universal protein knowledgebase in 2025. *Nucleic Acids Research*, 53:D609 – D617, 2024. 16

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

S. K. S. Chu and K. Y. Wei. Generative antibody design for complementary chain pairing sequences through encoder-decoder language model. *ArXiv*, abs/2301.02748, 2023. 9

T. Cohen and D. Schneidman-Duhovny. Epitope-specific antibody design using diffusion models on the latent space of ESM embeddings. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. 7

J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. 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. 9

S. Dieleman, L. Sartran, A. Roshannai, N. Savinov, Y. Ganin, P. H. Richemond, A. Doucet, R. Strudel, C. Dyer, C. Durkan, C. Hawthorne, R. Leblond, W. Grathwohl, and J. Adler. Continuous diffusion for categorical data. *ArXiv*, abs/2211.15089, 2022. 1, 27, 28

F. A. Dreyer, D. Cutting, C. Schneider, H. Kenlay, and C. M. Deane. Inverse folding for antibody sequence design using deep learning, 2023. URL https://arxiv.org/abs/2310.19513. 9, 15

J. Dunbar and C. M. Deane. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32:298 – 300, 2015. 17

J. Dunbar, K. Krawczyk, J. Leem, T. Baker, A. Fuchs, G. Georges, J. Shi, and C. M. Deane. 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. 15

A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. B. Fehér, C. Angerer, M. Steinegger, D. Bhowmik, and B. Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020. 15

N. C. Frey, D. Berenberg, K. Zadorozhny, J. Kleinhenz, J. Lafrance-Vanasse, I. Hotzel, Y. Wu, S. Ra, R. Bonneau, K. Cho, A. Loukas, V. Gligorijević, and S. Saremi. Protein discovery with discrete walk-jump sampling. *ArXiv*, abs/2306.12360, 2023. 2

T. Gao, X. Yao, and D. Chen. Simcse: Simple contrastive learning of sentence embeddings. *ArXiv*, abs/2104.08821, 2021. 5

M. Greenig, H. Zhao, V. Radenkovic, A. Ramon, and P. Sormanni. Igcraft: A versatile sequence generation framework for antibody discovery and engineering. *ArXiv*, abs/2503.19821, 2025. 7, 9

A. Greenshields-Watson, P. Agarwal, S. A. Robinson, B. H. Williams, G. L. Gordon, H. L. Capel, Y. Li, F. C. Spoendlin, B. Aguilar-Sanjuan, F. Boyles, and C. M. Deane. Anarcii: A generalised language model for antigen receptor numbering. *bioRxiv*, 2025. 15

B. Guloglu, M. Bragança, A. Graves, S. Cameron, T. Atkinson, L. Copoiu, A. Laterre, and T. D. Barrett. Abbfn2: A flexible antibody foundation model based on bayesian flow networks. *bioRxiv*, 2025. 7, 8

D. Guo, D. K. Dunn-Walters, F. Fraternali, and J. C. F. Ng. Immunomatch learns and predicts cognate pairing of heavy and light immunoglobulin chains. *Nature Methods*, 23:106 – 117, 2025. 9, 17

T. Hayes, R. Rao, H. Akin, N. J. Sofroniew, D. Oktay, Z. Lin, R. Verkuil, V. Q. Tran, J. Deaton, M. Wiggert, R. Badkundri, I. Shafkat, J. Gong, A. Derry, R. S. Molina, N. Thomas, Y. A. Khan, C. Mishra, C. Kim, L. J. Bartie, M. Nemeth, P. D. Hsu, T. Sercu, S. Candido, and A. Rives. Simulating 500 million years of evolution with a language model. *Science*, page eads0018, 2025. 7, 8, 9

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 20

E. Hoogeboom, D. Nielsen, P. Jaini, P. Forr'e, and M. Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Neural Information Processing Systems*, 2021. 7

C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives. 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. 9

M. H. Høie, A. M. Hummer, T. H. Olsen, B. Aguilar-Sanjuan, M. Nielsen, and C. M. Deane. 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. 9, 15

H. Kenlay, F. A. Dreyer, A. Kovaltsuk, D. Miketa, D. E. V. Pires, and C. M. Deane. Large scale paired antibody language models. *PLOS Computational Biology*, 20, 2024. 8

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

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

M.-P. Lefranc, C. Pommié, M. Ruiz, V. Giudicelli, E. Foulquier, L. D. Truong, V. Thouvenin-Contet, and G. Lefranc. 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, 15, 17

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. 2, 8

S. Luo, Y. Su, X. Peng, S. Wang, J. Peng, and J. Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 9754–9767. Curran Associates, Inc., 2022. 7

V. Meshchaninov, E. Chimbulatov, A. Shabalin, A. Abramov, and D. Vetrov. Compressed and smooth latent space for text diffusion modeling. *ArXiv*, abs/2506.21170, 2025a. 2

V. Meshchaninov, P. V. Strashnov, A. Shevtsov, F. Nikolaev, N. V. Ivanisenko, O. L. Kardymon, and D. Vetrov. Diffusion on language model encodings for protein sequence generation. In *International Conference on Machine Learning*, 2025b. 1, 7, 20

E. Nijkamp, J. A. Ruffolo, E. N. Weinstein, N. V. Naik, and A. Madani. Progen2: Exploring the boundaries of protein language models. *Cell systems*, 2022. 6, 15

T. H. Olsen, F. Boyles, and C. M. Deane. 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. 15

T. H. Olsen, I. H. Moal, and C. M. Deane. Ablang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2, 2022. 15

T. H. Olsen, I. H. Moal, and C. M. Deane. Addressing the antibody germline bias and its effect on language models for improved antibody design. *Bioinformatics*, 40, 2024. 8

M. I. J. Raybould, C. Marks, K. Krawczyk, B. Taddese, J. Nowak, A. P. Lewis, A. Bujotzek, J. Shi, and C. M. Deane. Five computational developability guidelines for therapeutic antibody profiling. *Proceedings of the National Academy of Sciences*, 116(10):4025–4030, Mar. 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1810576116. URL https://pnas.org/doi/full/10.1073/pnas.1810576116. 28

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

J. A. Ruffolo, J. J. Gray, and J. Sulam. Deciphering antibody affinity maturation with language models and weakly supervised learning. *ArXiv*, abs/2112.07782, 2021. 8

J. A. Ruffolo, L.-S. Chu, S. P. Mahajan, and J. J. Gray. Fast, accurate antibody structure prediction from deep learning on massive set of natural antibodies. *Nature Communications*, 14(1):2389, Apr. 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38063-x. URL https://www.nature.com/articles/s41467-023-38063-x. 9, 34

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

H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *ArXiv*, abs/2006.07710, 2020. 3

R. W. Shuai, J. A. Ruffolo, and J. J. Gray. Iglm: Infilling language modeling for antibody sequence design. *Cell systems*, 2023. 7, 8, 15

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

M. Steinegger and J. Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35:1026–1028, 2017. 16

P. Strashnov, A. Shevtsov, V. Meshchaninov, O. Kardymon, and D. Vetrov. 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. 34

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

O. M. Turnbull, D. Oglic, R. Croasdale-Wood, and C. M. Deane. p-iggen: a paired antibody generative language model. *Bioinformatics*, 40, 2024. 7, 9

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

J. Yim, B. L. Trippe, V. D. Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. Se(3) diffusion model with application to protein backbone generation. *ArXiv*, abs/2302.02277, 2023. 16

Appendix

Contents

A Dataset **15**
- A.1 Unpaired Dataset . . . 15
- A.2 Paired Dataset . . . 15
- A.3 Structure Dataset . . . 15

B Evaluation metrics **15**
- B.1 Distributional Similarity . . . 15
- B.2 Sequence Plausibility . . . 15
- B.3 Diversity . . . 16
- B.4 Diversity Score for CDR Infilling . . . 16
- B.5 Amino Acid Recovery (AAR) . . . 16
- B.6 Edit Distance. . . . 17
- B.7 IM-score. . . . 17
- B.8 Structure-Based Metrics . . . 18

C Architecture Details **19**
- C.1 Autoencoder Architecture . . . 19
- C.2 Length Handling and Normalization . . . 19
- C.3 Diffusion Denoiser . . . 20

D Training Details **20**
- D.1 Pretraining . . . 20
- D.2 Finetuning on infilling task . . . 20
- D.3 Finetuning on pairing task . . . 21
- D.4 Finetuning on structure conditioning task . . . 21
- D.5 Compute Resources . . . 21
- D.6 Inference Timing . . . 21

E Additional Experiments **22**
- E.1 RLD Generalization to Different Encoder Models . . . 22
- E.2 Impact of Autoencoder Dimensions . . . 23
- E.3 Autoencoder Ablation . . . 25
- E.4 CDR3 Reconstruction under Latent Corruption . . . 25
- E.5 Latent Space Smoothness Evaluation . . . 26
- E.6 Per-Position Loss Reweighting and Noise Schedules . . . 27
- E.7 Fréchet Distance across Evaluation Embedding Spaces . . . 28
- E.8 Computational Developability Assessment . . . 28

F Organism-Conditioned Generation **30**

G Sequence Infilling **31**
- G.1 Variable-Length CDR3 Infilling . . . 31
- G.2 Single-Span Fixed-Length CDR3 Infilling . . . 32
- G.3 Multi-Span Fixed-Length All-CDR Infilling . . . 33
- G.4 Implications . . . 33

H Structure-Conditioned Generation **34**
- H.1 Test-Set Construction . . . 34
- H.2 RMSD Results . . . 34
- H.3 Per-Region Amino-Acid Recovery . . . 34

I Broader Impacts **35**

J Safeguards **35**

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).

We release the exact filtered dataset at HuggingFace.

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 and 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 infilling task, we measure diversity as the mean normalized pairwise distance among the K generated CDR sequences, retaining duplicate samples. Higher values indicate greater variation among generated sequences, while identical outputs yield zero diversity. For variable-length sequences, we use Levenshtein distance normalized by the maximum length of the two sequences. For aligned fixed-length sequences, we use Hamming distance normalized by sequence length.

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 and 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.

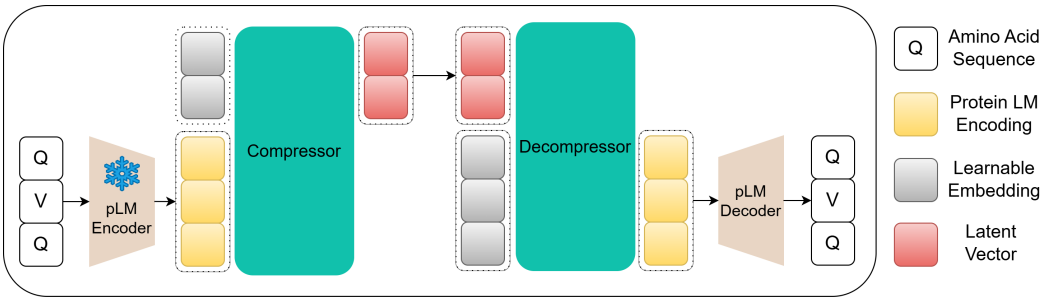


Figure 8: **Compressor–decompressor architecture.** The compressor concatenates the frozen PLM activations $h \in \mathbb{R}^{L\times d_{\text{enc}}}$ with N learnable query vectors $u \in \mathbb{R}^{N\times d}$ and processes the joint stack through a 12-layer cross-attention transformer; the latent $z \in \mathbb{R}^{N\times d}$ 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, both LDM encoder representations and RLD redistributed latents are normalized by per-dimension z-scoring, with mean and variance estimated on a held-out subset of training sequences. This normalization 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. For each encoder, LDM and RLD use the same denoiser backbone. Their input and output projections accommodate the respective representation dimensions.

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 pretrain on the filtered unpaired dataset described in Sec. A.1 for up to 1M steps, using a batch size of 512 and a learning-rate schedule with warmup followed by cosine decay, with minimum and maximum learning rates of 10^{-4} and 4×10^{-4}, respectively. We condition on organism and chain type during pretraining, enabling conditional generation without additional finetuning. Each conditioning label is independently dropped with probability 0.1 (see Sec. F).

Within each matched LDM–RLD comparison, diffusion training uses the same number of steps, batch size, and optimizer settings. RLD additionally requires autoencoder training; the autoencoder is frozen during diffusion training. We keep the pretrained protein encoder frozen in both settings, following DiMA, to isolate the effect of representation redistribution from encoder adaptation. Joint encoder finetuning is not evaluated.

D.2 Finetuning on infilling task

We finetune on the same filtered unpaired dataset used for pretraining. We initialize from an RLD checkpoint pretrained for 1M steps using ESM3-based redistributed latents with $N = 16$ and $d = 512$. For each training sequence, we mask either CDR3 alone or all three CDRs and encode the masked sequence using the same ESM3 encoder and RLD compressor to obtain the conditioning representation C_0. A linear projection maps C_0 to the diffusion hidden dimension, and each score-estimator transformer block accesses it through cross-attention.

We finetune the diffusion model and conditioning projection for 200k steps with a batch size of 512, learning-rate warmup, and a maximum learning rate of 2×10^{-4}. For model comparisons, we use 100 randomly selected antibody pairs from the held-out 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. Autoencoder training (100k steps, batch size 512) takes approximately 3 hours on 8 A100s. Diffusion pretraining (1M steps, batch size 512) takes approximately 40 hours on 8 A100s per encoder configuration. Each conditional finetuning run for 100k–200k steps (infilling, pairing, structure-conditioning) requires additional 8 hours on 8 A100s.

D.6 Inference Timing

We measure wall-clock generation time for 2,048 human heavy or light chain sequences, averaged over five runs. LDM and RLD use 200 diffusion sampling steps, and RLD uses $N = 128$ latent slots with feature dimension $d = 512$. We compare both diffusion models on one and eight A100 GPUs and include the autoregressive IgLM baseline on one GPU.

Table 7: **Inference timing.** Wall-clock time in seconds to generate 2,048 human heavy or light chain sequences. Values represent the mean and standard deviation across five runs. Both diffusion models use 200 sampling steps. RLD uses $N = 128$ and $d = 512$. Lower is better, fastest times for each GPU count are bold.

GPUs	Model	Generation time (s) ↓ Heavy	Light
1	LDM	85.775 ± 0.271	66.922 ± 0.204
1	RLD	**61.948** ± 0.163	**61.861** ± 0.007
1	IgLM	715.946 ± 20.323	560.642 ± 18.254
8	LDM	11.655 ± 0.123	8.926 ± 0.011
8	RLD	**8.313** ± 0.132	**8.245** ± 0.004

RLD generates 2,048 sequences in approximately 62 seconds on one GPU and 8.3 seconds on eight GPUs. Across the tested settings, it is approximately 1.1–1.4 times faster than LDM; on one GPU, it is approximately 9–12 times faster than IgLM. RLD requires one decompressor pass after denoising, while its fixed latent length reduces the number of positions processed at each diffusion step. Generation times are nearly identical for heavy and light chains.

E Additional Experiments

E.1 RLD Generalization to Different Encoder Models

We test generalization of our method on 5 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. 8 and Fig. 9.

Table 8: **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 $\pm$ 0.01	0.25 $\pm$ 0.01	2.62 $\pm$ 0.01	2.75 $\pm$ 0.01	0.971 $\pm$ 0.005	0.839 $\pm$ 0.005
ESM-2	LDM	6.15 $\pm$ 1.48	4.60 $\pm$ 0.50	1.57 $\pm$ 0.01	1.59 $\pm$ 0.02	0.283 $\pm$ 0.014	0.028 $\pm$ 0.005
	RLD	0.57 $\pm$ 0.02	0.61 $\pm$ 0.03	2.30 $\pm$ 0.01	2.40 $\pm$ 0.01	0.978 $\pm$ 0.005	0.789 $\pm$ 0.007
ESM3	LDM	9.02 $\pm$ 0.80	5.71 $\pm$ 0.50	1.79 $\pm$ 0.02	1.82 $\pm$ 0.02	0.157 $\pm$ 0.015	0.037 $\pm$ 0.005
	RLD	0.49 $\pm$ 0.02	0.49 $\pm$ 0.01	2.29 $\pm$ 0.01	2.40 $\pm$ 0.01	0.957 $\pm$ 0.008	0.749 $\pm$ 0.010
IgBert	LDM	6.04 $\pm$ 0.60	4.51 $\pm$ 0.45	1.63 $\pm$ 0.02	1.60 $\pm$ 0.02	0.346 $\pm$ 0.030	0.031 $\pm$ 0.005
	RLD	0.46 $\pm$ 0.02	0.39 $\pm$ 0.01	2.39 $\pm$ 0.02	2.50 $\pm$ 0.02	0.973 $\pm$ 0.004	0.782 $\pm$ 0.023
AntiBERTy	LDM	5.38 $\pm$ 0.50	4.39 $\pm$ 0.40	1.73 $\pm$ 0.02	1.61 $\pm$ 0.02	0.679 $\pm$ 0.050	0.038 $\pm$ 0.005
	RLD	0.47 $\pm$ 0.02	0.65 $\pm$ 0.03	2.38 $\pm$ 0.02	2.32 $\pm$ 0.02	0.978 $\pm$ 0.006	0.658 $\pm$ 0.015
AbLang2	LDM	8.28 $\pm$ 0.80	6.22 $\pm$ 0.60	2.00 $\pm$ 0.02	2.14 $\pm$ 0.02	0.405 $\pm$ 0.040	0.066 $\pm$ 0.010
	RLD	0.39 $\pm$ 0.01	0.47 $\pm$ 0.21	2.42 $\pm$ 0.01	2.57 $\pm$ 0.06	0.976 $\pm$ 0.012	0.800 $\pm$ 0.013

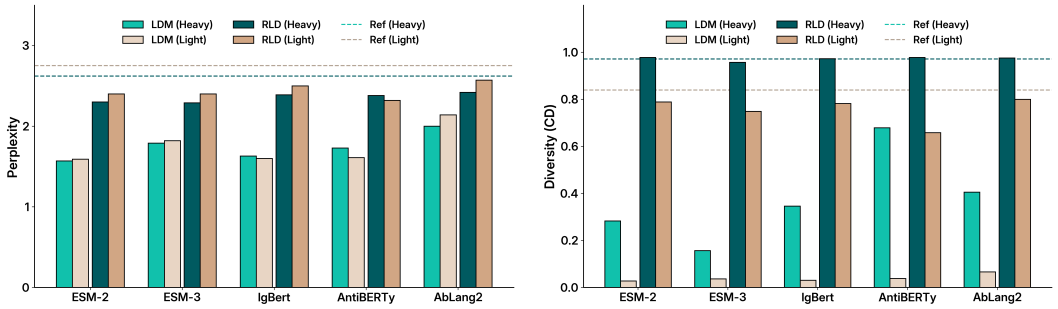


Figure 9: **Latent redistribution fixes capacity misallocation regardless of encoder architecture.**

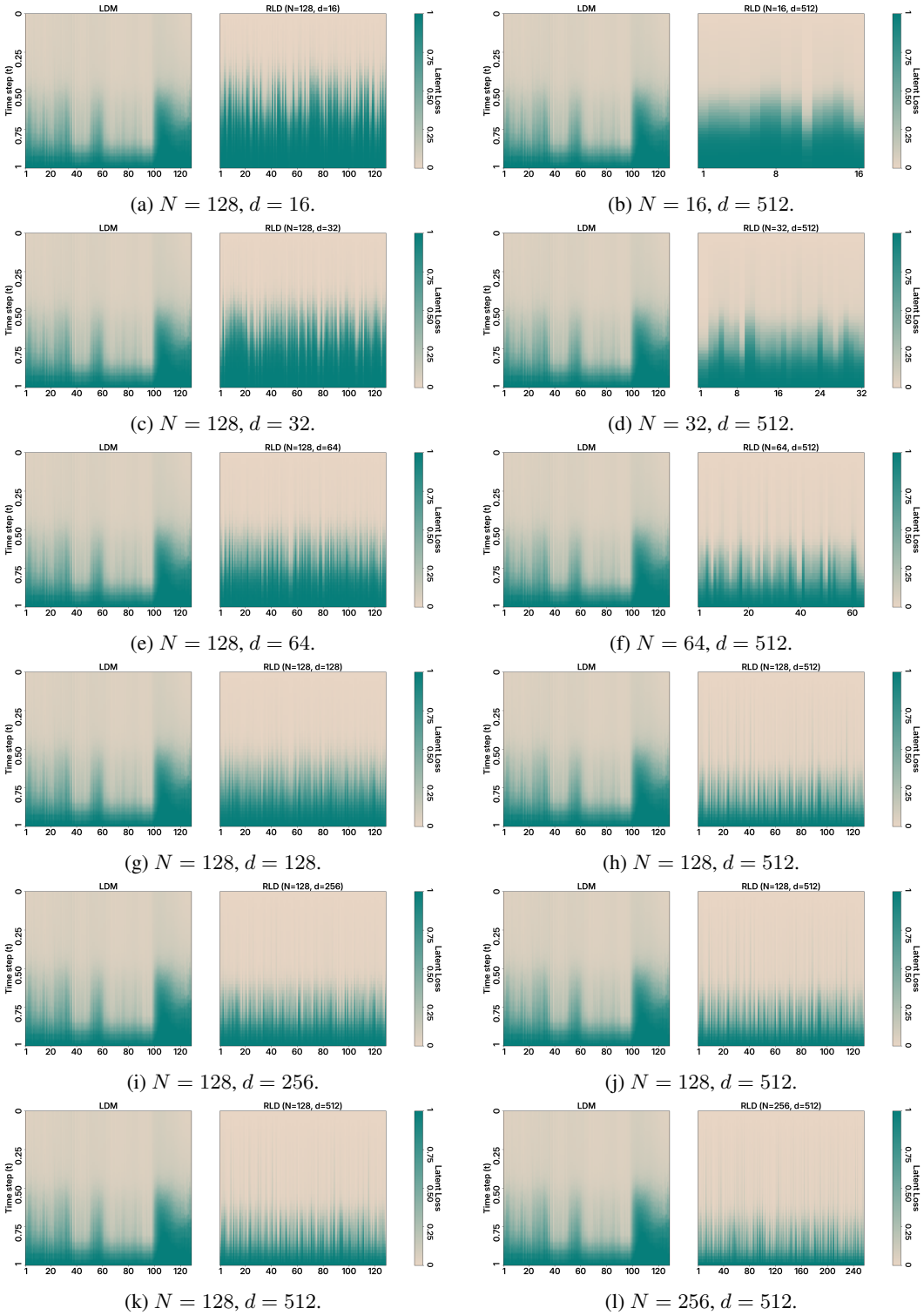
E.2 Impact of Autoencoder Dimensions

We vary the latent feature dimension d and slot count N to evaluate their effect on generation quality. For each configuration in Tab. 9, we train the autoencoder for 100k steps and the diffusion model for 1M steps using ESM-2 150M representations.

Table 9: **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. 9 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

Training components. We evaluate the autoencoder training components using ESM-2 representations. We train both the autoencoder and diffusion model for 100k steps, comparing reconstruction objectives alone, input augmentations, and input augmentations combined with latent dropout. Table 10 reports generation metrics, diffusion loss standard deviation, and latent-space smoothness for human heavy and light chains.

Table 10: **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

Latent size and regularization. To test whether redistribution requires a dimensional bottleneck, we compare $N = 128$ and $N = 256$ with $d = 640$, matching the input dimension of the encoder (ESM-2 150M). For each latent size, we evaluate the same three training configurations. Both the autoencoder and diffusion model are trained for 100k steps. Since inputs are padded to $L = 154$, the $N = 256$ configuration is overcomplete along the sequence axis. Table 11 reports the results.

Table 11: **Latent size and regularization ablation.** Evaluation of FD, Perplexity (PPL), and CD across human heavy and light chains for two latent-slot counts and three training configurations. Each RLD configuration uses 100k autoencoder training steps and 100k diffusion training steps. Dataset and LDM values are reproduced from Table 10 for reference.

Model	N	FD ↓ (10^{-2})		PPL		CD ↑	
		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
RLD	128	0.92 ± 0.07	0.91 ± 0.02	2.18 ± 0.01	2.29 ± 0.02	0.976 ± 0.006	0.754 ± 0.003
RLD + aug.	128	0.74 ± 0.05	0.78 ± 0.22	2.25 ± 0.02	2.41 ± 0.09	0.914 ± 0.015	0.752 ± 0.018
RLD + aug. + drop	128	0.59 ± 0.06	0.56 ± 0.03	2.33 ± 0.03	2.44 ± 0.01	0.979 ± 0.006	0.759 ± 0.012
RLD	256	1.08 ± 0.03	1.42 ± 0.09	2.18 ± 0.01	2.25 ± 0.01	0.971 ± 0.005	0.725 ± 0.019
RLD + aug.	256	0.72 ± 0.04	0.76 ± 0.04	2.26 ± 0.02	2.38 ± 0.02	0.971 ± 0.006	0.720 ± 0.019
RLD + aug. + drop	256	0.53 ± 0.02	0.57 ± 0.04	2.37 ± 0.02	2.41 ± 0.02	0.977 ± 0.005	0.751 ± 0.018

RLD substantially improves FD and diversity over LDM at both latent sizes, even without input augmentations or latent dropout. Adding augmentations and then dropout further reduces FD on both chains at both sizes. The gains persist in the overcomplete configuration, showing that a sequence-axis bottleneck is not necessary. These results support the interpretation that cross-attention enables redistribution across learned latent slots, and regularization further improves generation quality.

E.4 CDR3 Reconstruction under Latent Corruption

We test whether CDR3 information remains recoverable when part of the representation is corrupted. Using the overcomplete configuration ($N = 256$, $d = 640$), we encode 512 held-out human heavy chains with each RLD autoencoder variant and compare against the original ESM-2 representations. We randomly replace 10%, 25%, 50%, or 75% of representation positions with random vectors drawn within the observed range of each channel, and include uncorrupted representations as a reference. We then decode the corrupted representations and measure CDR3 amino-acid recovery, using 100 random corruption draws per level.

Table 12 shows perfect CDR3 reconstruction for all configurations without corruption. At every nonzero corruption level, RLD retains higher CDR3 recovery than ESM-2, even without input augmentations or latent dropout. Both training components further improve recovery. With half

Table 12: **CDR3 reconstruction under latent corruption.** CDR3 amino-acid recovery on 512 held-out human heavy chains after randomly replacing a fraction of representation positions. RLD variants use the overcomplete configuration ($N = 256$, $d = 640$). We evaluate 100 random corruption draws per nonzero corruption level. Higher recovery is better; best values in each row are bold.

Corrupted positions	RLD	RLD + aug.	RLD + aug. + drop	ESM-2
0%	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00
10%	0.96 ± 0.06	0.97 ± 0.05	**0.98** ± 0.03	0.91 ± 0.08
25%	0.86 ± 0.09	0.91 ± 0.10	**0.95** ± 0.06	0.76 ± 0.11
50%	0.67 ± 0.14	0.74 ± 0.16	**0.84** ± 0.11	0.57 ± 0.13
75%	0.37 ± 0.16	0.46 ± 0.17	**0.55** ± 0.19	0.32 ± 0.12

the representation positions corrupted, the full model recovers 84% of CDR3 residues, compared with 57% for ESM-2; at 75% corruption, recovery remains 55% versus 32%. These results are consistent with CDR3 information being distributed more broadly across RLD latent slots and show that regularization strengthens robustness to partial corruption.

E.5 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. These experiments use 100k autoencoder training steps and 100k diffusion training steps. 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 13: **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 14: **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.6 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. 8. 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: $0.283 \rightarrow 0.566$; CD L: $0.028 \rightarrow 0.746$) 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.

Table 15: **Process-level alternatives on human heavy and light chains.** FD values are scaled by 10^{-2}. Reported values are mean ± 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})		CD ↑	
	Heavy	Light	Heavy	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

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 follows the time-warping framework of Dieleman et al. [2022]], extended here to per-position warping, with their EMA coefficient of 0.9 retained. The remaining choices were guided by preliminary runs and statistical considerations: the 4-gram window balances local context and estimation reliability, the weight-ratio cap avoids the instability observed with uncapped weights, and the warp grid preserves the empirical loss CDFs. More tuning could narrow the gap between RLD and the adaptive schedule. The structural distinction remains: both process-level methods require statistics defined on IMGT-aligned positions, whereas RLD learns redistribution without position-specific weighting or scheduling rules. The same RLD training recipe also transfers across the five tested encoders.

E.7 Fréchet Distance across Evaluation Embedding Spaces

To assess whether the FD improvement depends on the evaluation embedding, we repeat the comparison in five additional embedding spaces: ESM-2 650M, ESM3, AbLang2, AntiBERTy, and IgBert. Both RLD and LDM were trained using ESM-2 150M representations. We use the same pooling procedure and FD estimator as in Sec. B.1, changing only the evaluation encoder. Since embedding scales differ, FD values should be compared within each embedding space.

Table 16: **FD across evaluation embedding spaces.** Comparison of RLD and LDM trained on ESM-2 150M representations, evaluated on human heavy (H) and light (L) chains using five alternative embedding models. Test denotes the natural test-set reference. ESM3 values are reported in units of 10^{-4}. Lower FD is better; best generated-model values for each chain and embedding space are bold.

Model	Chain	ESM-2 650M ↓	ESM3 ↓ (10^{-4})	AbLang2 ↓	AntiBERTy ↓	IgBert ↓
Test	H	0.014 ± 0.001	0.27 ± 0.05	0.068 ± 0.004	2.91 ± 0.04	0.015 ± 0.001
RLD	H	**0.041** ± 0.005	**6.17** ± 1.13	**0.228** ± 0.011	**9.02** ± 0.21	**0.138** ± 0.009
LDM	H	0.503 ± 0.019	126.40 ± 5.50	6.12 ± 0.11	78.17 ± 1.86	1.48 ± 0.04
Test	L	0.013 ± 0.001	0.38 ± 0.10	0.056 ± 0.002	0.390 ± 0.014	0.014 ± 0.000
RLD	L	**0.025** ± 0.001	**5.38** ± 0.53	**0.324** ± 0.020	**0.514** ± 0.022	**0.145** ± 0.007
LDM	L	0.183 ± 0.005	71.04 ± 2.07	3.49 ± 0.04	3.16 ± 0.11	1.57 ± 0.02

RLD achieves lower FD than LDM in all five evaluation spaces on both heavy and light chains, reproducing the ordering observed with ProtT5. The reductions range from approximately 6-fold to 27-fold. These results indicate that the improvement in distributional similarity is not specific to the ProtT5 evaluation space.

E.8 Computational Developability Assessment

We assess generated sequences using TAP-style developability metrics Raybould et al. [2019] computed from IgFold-predicted structures. We evaluate hydrophobic surface patches (PSH), positive and negative charge patches (PPC and PNC), and net charge for human heavy and light chains. For each model, we evaluate 128 sequences per run over four runs. We compare RLD and LDM with natural test sequences and include sequences with randomized CDRs as a negative control.

We assess agreement with the natural test distribution rather than treating higher or lower metric values as universally preferable. Table 17 reports metric values and two-sided Kolmogorov–Smirnov (KS) test p-values against the natural test set.

Table 17: **Computational developability assessment.** TAP-style metrics computed from IgFold-predicted structures for human heavy (H) and light (L) chains. Rand. CDR denotes the randomized-CDR control. The final three columns report two-sided KS test p-values against the natural test distribution. Values above 0.05 indicate that no statistically significant difference is detected at the nominal threshold; they do not establish equivalence.

Metric	**Chain**	Metric values				KS p-value vs. test		
		Test	RLD	LDM	Rand. CDR	RLD	LDM	Rand. CDR
PSH	H	67.61 $\pm$ 0.52	67.50 $\pm$ 0.54	66.11 $\pm$ 0.65	69.52 $\pm$ 0.44	0.964	0.005	0.002
PSH	L	51.23 $\pm$ 0.75	50.80 $\pm$ 1.14	50.88 $\pm$ 0.39	52.59 $\pm$ 0.98	0.475	0.002	1.8×10^{-5}
PPC	H	0.106 $\pm$ 0.016	0.125 $\pm$ 0.020	0.069 $\pm$ 0.007	0.234 $\pm$ 0.048	0.680	4.0×10^{-4}	2.8×10^{-8}
PPC	L	0.124 $\pm$ 0.010	0.150 $\pm$ 0.033	0.087 $\pm$ 0.013	0.231 $\pm$ 0.015	0.628	1.9×10^{-8}	1.7×10^{-24}
PNC	H	0.141 $\pm$ 0.005	0.177 $\pm$ 0.057	0.095 $\pm$ 0.016	0.208 $\pm$ 0.016	0.575	0.054	0.003
PNC	L	0.067 $\pm$ 0.026	0.066 $\pm$ 0.015	0.034 $\pm$ 0.019	0.150 $\pm$ 0.038	1.000	0.001	1.3×10^{-8}
Net charge	H	2.279 $\pm$ 0.142	2.120 $\pm$ 0.296	2.277 $\pm$ 0.175	2.839 $\pm$ 0.152	0.343	0.032	8.8×10^{-9}
Net charge	L	0.527 $\pm$ 0.202	0.500 $\pm$ 0.262	0.635 $\pm$ 0.100	1.004 $\pm$ 0.155	0.575	0.001	2.5×10^{-5}

At the nominal significance threshold of 0.05, no statistically significant difference from the natural test distribution is detected for RLD on any of the eight metric–chain combinations. Differences are detected for LDM on seven of eight combinations and for the randomized-CDR control on all eight. These results provide computational evidence that RLD preserves the evaluated developability-related properties of natural antibodies. Nonsignificant KS results do not establish distributional equivalence, and these computational assessments do not demonstrate experimental stability or antigen binding.

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 18: **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 19: **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.

Table 20: **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

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.

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 21 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 21: **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

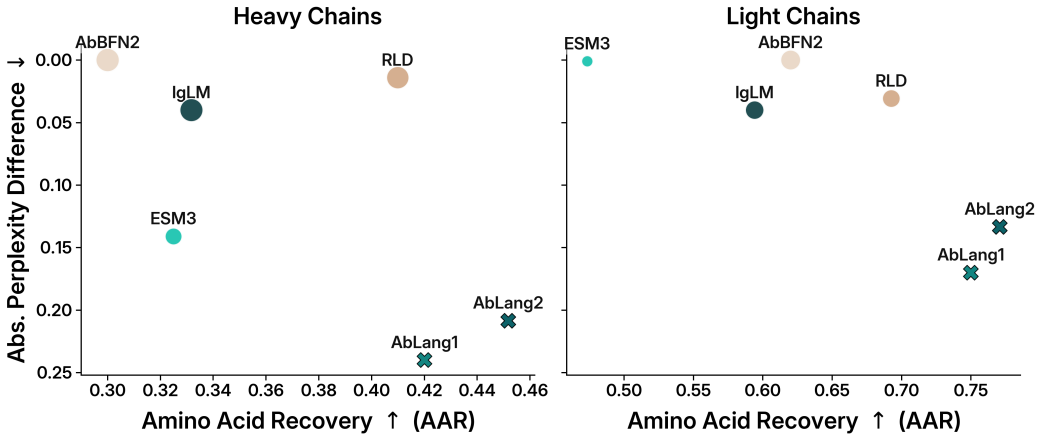


Figure 11: **CDR3 infilling performance across heavy and light chains.** Models regenerate masked CDR3 regions conditioned on framework context. This task presents a Pareto trade-off between **fidelity** (Amino Acid Recovery, higher is better) and **generation quality** (Absolute Perplexity Difference 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 22 shows detailed CDR3 infilling results. RLD achieves 40.8% AAR on heavy chains and 68.9% AAR on light chains, with perplexity differences below 0.03 and diversity scores of 0.77 and 0.52, respectively. This combination indicates that RLD balances reconstruction fidelity, sequence naturalness, and sample diversity.

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 (41.9–45.2% on heavy chains and 75.2–77.1% on light chains) but use deterministic argmax decoding and exhibit elevated perplexity differences (0.13–0.25). These results illustrate a trade-off between reconstruction fidelity and sequence naturalness.

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 22: **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.**

Model	AAR ↑		Diversity ↑		Exact match ↓		|ΔPPL| ↓	
	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 23 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 23: **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.**

Model	AAR ↑		Diversity ↑		Exact match ↓		|ΔPPL| ↓	
	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	2.719 ± 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	0.030 ± 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	0.023 ± 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.

We make the testing set available at HuggingFace

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: RLD achieves the lowest global and CDR3 RMSD on light chains, while AntiFold achieves the lowest values on heavy chains. AbMPNN outperforms ESM3 on all four metrics. The general-protein inverse folding methods, ESM-IF1 and ProteinMPNN, show the largest errors on heavy chains, with CDR3 RMSD values of 5.07 Åand 3.86 Å, respectively.

H.3 Per-Region Amino-Acid Recovery

Tab. 24 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 24: **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

I 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, since our model is trained on natural antibody repertoires from OAS and it generates sequences conditioned on organism, chain type, framework context, or backbone structure. The evaluated model does not condition on antigen identity or perform affinity maturation. Extending its conditioning pathway to antigen or epitope information would require additional training and validation, and its potential benefits and misuse risks would need to be reassessed. More generally, improved generative tooling for biomolecules may lower barriers to misuse over time.

J Safeguards

We assess the misuse risk of the released model and data as low, and adopt proportionate safeguards. The model generates antibody variable-domain sequences conditioned on organism, chain type, surrounding framework context, or backbone structure. It does not condition on antigen identity, does not perform affinity maturation, and does not predict binding to any specified target. Translating a generated sequence into a functional therapeutic requires antigen-specific design, expression, purification, and experimental characterization, none of which the model provides. The training data (OAS, SAbDab) is publicly available and contains only natural antibody repertoires from healthy donors and deposited structures; we do not introduce new sensitive data. Released artifacts (filtered training subset, structure test set, model weights, source code) are distributed under the upstream licenses listed above, with documentation describing intended use for methodological research on generative modelling of antibody sequences. We do not implement gated access, as the capabilities provided are comparable to existing publicly released antibody language models (IgLM, AbLang2, AbBFN2, IgCraft, p-IgGen) and do not extend the misuse frontier. Should follow-up work extend RLD to antigen-conditioned design or affinity optimization, we recommend revisiting this assessment.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction claims (10× FD reduction over LDM, encoder-agnostic redistribution, support for chain-type/CDR/paired/structure-conditioned generation) are all backed by Sec. 5.2 (six organisms), Sec. 5.3 (five encoders), Secs. 5.4 to 5.6 (conditional tasks).

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations are discussed in context: the heavy-vs-light gap (Sec. G.2), the AAR–naturalness trade-off (Secs. 5.4 and G.2), the saturation of structural fidelity metrics on antibodies (Sec. B.8), and the fact that more tuning could narrow the gap with adaptive noise schedules (Sec. E.6).

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper formalizes the SNR/global-schedule mismatch but does not state formal theorems requiring proofs.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Architecture (Sec. C), training recipes for pretraining and all three finetuning tasks (Sec. D), evaluation protocols (Sec. B), and dataset construction (Sec. A) are described. Source code and the datasets are released.

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

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

   Answer: [Yes]

   Justification: The source code and datasets are released. The links are at the end of introduction and in Sec. A.1 and Sec. H.1.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Optimizer schedule (warmup + cosine), learning rates (max 4e-4 pretrain, 2e-4 finetune), batch size 512, training steps (1M diffusion pretraining, 100k–200k task finetuning), latent dimensions ($N = 128$, $d = 16$), and condition-drop probability (0.1) are specified in Sec. D

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Tables 1, 7–18 report mean ± standard deviation across five independent runs of 2,048 generated sequences. The variability source (independent generation runs) is stated.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The information on the computer resources is discussed in Sec. D.5

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses public datasets (OAS, SAbDab) and standard ML protocols, with no human-subjects component.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Broader impacts are discussed in Sec. I

11. **Safeguards**

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

    Answer: [Yes]

    Justification: The safeguards are discussed in Sec. J

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All assets (OAS, SAbDab, ESM-2, ESM3, ProGen2, IgFold, AntiFold, AbMPNN, ProteinMPNN, ESM-IF1, IgLM, AbBFN2, IgCraft, p-IgGen, pAbT5, MMseqs2, ProtT5, ANARCII) are cited.

13. **New assets**

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

    Answer: [Yes]

    Justification: The filtered OAS subset, the structure-conditioned test set, and the source code are released via HuggingFace and GitHub. Documentation accompanies them.

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

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

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects.

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

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

Answer: [N/A]

Justification: No human-subjects research.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs are not used as a non-standard component of the core method.