MOLECULES MEET LANGUAGE: CONFOUND-AWARE REPRESENTATION LEARNING AND CHEMICAL PROPERTY STEERING IN TRANSFORMER-VAE LATENT SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular generative models often assume meaningful latent geometry, but apparent property predictability can reflect sequence-level shortcuts rather than chemical organization. We study this issue in an unsupervised autoregressive Transformer-VAE trained on SELFIES. After training, we freeze the model, fit linear probes to RDKit descriptors, and use the probe weights as candidate global steering directions. To separate chemical signal from SELFIES artifacts, we introduce a confound-aware evaluation based on residualization, confound-direction alignment analysis, and decoded-molecule traversal. This is necessary because SELFIES length, branch tokens, ring tokens, and token entropy are strongly encoded in the latent space. Under this confound-aware evaluation, we find robust monotonic steering for cLogP, FractionCSP3, HeavyAtomCount, TPSA, BertzCT, and HBA. Nonlinear probes further show that some properties admit stable global directions, while others are better described by local latent gradients. Overall, our results show that chemically meaningful steering can emerge in entangled molecular latent spaces, but only when validated through decoded molecules and controlled for representation-level confounds.

1 INTRODUCTION

Molecular design is a central problem in drug discovery and materials science, where the objective is to generate or optimize chemical structures with desired properties Sanchez-Lengeling & Aspuru-Guzik (2018); Vamathevan et al. (2019); Du et al. (2022). Although molecules are naturally represented as labeled graphs, molecular line notations make them accessible to sequence models. SMILES provides a compact chemical language for molecular graphs Weininger (1988), and its non-unique forms have been used as data augmentation in molecular learning Bjerrum (2017). SELFIES addresses a different but crucial issue for generation: arbitrary SELFIES strings decode to valid molecules under the representation grammar Krenn et al. (2020). In this work, molecular language is therefore used as the training interface, while chemical validity and molecular properties are evaluated after decoding into graph-based molecular objects.

Language modeling has become a general mechanism for learning representations from unlabeled sequences. Transformers enabled scalable sequence modeling through attention Vaswani et al. (2017), while large-scale pretraining demonstrated that useful representations can emerge without task-specific supervision Devlin et al. (2019). Related principles have been observed in biological sequences Rives et al. (2021) and chemical strings, including SMILES Transformer fingerprints Honda et al. (2019), ChemBERTa-style molecular pretraining Chithrananda et al. (2020), SELFIES-based Transformer representations Yuksel et al. (2023), and autoregressive molecular generation with MolGPT Bagal et al. (2021). These works support the view that molecular strings contain learnable chemical regularities. However, they do not fully answer whether an unsupervised molecular language model learns a continuous latent space with chemically meaningful and steerable directions.

Molecular generation and optimization have followed complementary directions. Optimization searches in discrete molecular graph or string space using reinforcement learning, genetic search, or

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

graph-based policies You et al. (2018); Zhou et al. (2019); Jensen (2019); Fu et al. (2022). Generative approaches instead learn a distribution over molecules and use the learned representation for sampling, interpolation, or optimization, including molecular VAEs, junction-tree models, normalizing flows, and graph diffusion models Gomez-Bombarelli et al. (2018); Jin et al. (2018); Zang & Wang (2020); Vignac et al. (2023). Latent-space optimization exploits trained generators as continuous search spaces, for example through Bayesian optimization, latent inceptionism, linear molecular manipulation, or learned latent flows Griffiths & Hernandez-Lobato (2020); Eckmann et al. (2022); Du et al. (2023); Wei et al. (2024). These methods show that latent molecular spaces can support optimization, in a nontrivial manner: a useful generator does not necessarily provide globally meaningful chemical directions.

This paper studies that question in a deliberately constrained setting. We train a slot-based autoregressive Transformer-VAE on SELFIES sequences using only reconstruction and latent regularization. After training, the encoder and decoder are frozen. We then fit linear probes to RDKit-computed molecular descriptors and use the learned probe weights as candidate global steering directions. Thus, property labels are not used to train the generative model; they are used only post hoc to interrogate the latent space. The main evaluation is not whether a descriptor can be predicted from a molecular string, but whether moving through the frozen latent space along a probe-derived direction changes decoded molecules in the intended chemical direction.

A central risk is that a molecular language model may encode sequence-level shortcuts rather than chemical structure. SELFIES improves validity, but it remains a token sequence whose length, branch tokens, ring tokens, and token entropy can correlate with molecular size, topology, and descriptor values. We therefore evaluate latent directions under a confound-aware protocol: descriptors must be predictable from the frozen latent representation, robust to measured SELFIES-level confounds, and validated after decoding through molecular properties rather than latent probe scores alone.

The full framework is summarized in Figure 1. Our contributions are as follows:

- We frame molecular property steering as post hoc interpretation of an unsupervised molecular language model trained on SELFIES.
- We connect linear probing to latent chemical control by treating probe weights as global steering axes and validating them through decoded molecules.
- We introduce a confound-aware evaluation that separates chemical organization from SELFIES-level shortcuts such as token length, branch-token count, ring-token count, and token entropy.
- We show that useful global directions can emerge in an entangled latent space, with cLogP, FractionCSP3, BertzCT, TPSA, and HBA exhibiting robust monotonic steering.
- We use nonlinear probes diagnostically to distinguish globally linear steerability from local or nonlinear latent organization.

2 Background

2.1 Molecular representations and chemical language

Molecules are graphs whose atoms and bonds define chemical structure. Graph-based generators model this structure directly, avoiding arbitrary traversal order, and have been developed through junction-tree decomposition, implicit graph generation, invertible graph flows, and discrete graph diffusion Jin et al. (2018); De Cao & Kipf (2018); Zang & Wang (2020); Vignac et al. (2023). Their advantage is structural explicitness; their difficulty is that valid molecular generation remains a constrained discrete problem involving rings, graph size, and graph isomorphism.

String representations expose the same chemical objects to sequence models. SMILES is compact, expressive, and widely supported, but valid generation requires learning a constrained formal language involving branches, ring closures, aromaticity, and stereochemical conventions Weininger (1988). Canonicalization can impose a deterministic form, while non-canonical SMILES can also be exploited as augmentation Bjerrum (2017). SELFIES changes this trade-off by embedding stronger validity constraints into the representation grammar, making it particularly suitable for sequence-based generative modeling Krenn et al. (2020).

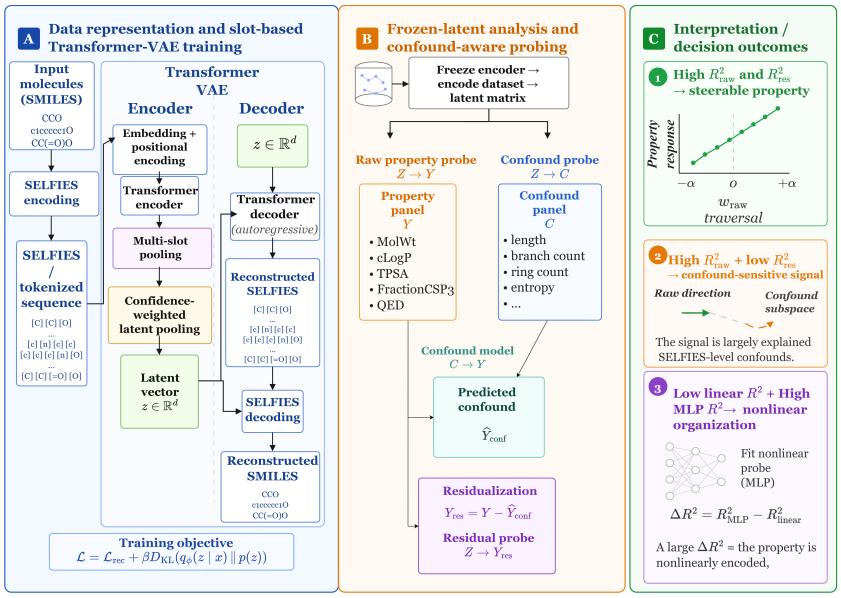

Figure 1: Our framework. We train an unsupervised autoregressive SELFIES VAE. We freeze the encoder and fit raw and confound-controlled property probes. We validate candidate directions after decoded-molecule traversal; residual directions provide an auxiliary confound-robustness diagnostic.

2.2 Chemical language models and latent molecular generation

Transformers provide a natural architecture for molecular strings because autoregressive decoding can model token sequences one symbol at a time Vaswani et al. (2017). More broadly, self-supervised language modeling has shown that useful structure can be recovered from unlabeled sequences in natural language and biology Devlin et al. (2019); Rives et al. (2021). Chemical language models extend this principle to molecular strings through unsupervised SMILES pretraining, large-scale molecular Transformers, SELFIES-based representation learning, and GPT-style molecular generation Honda et al. (2019); Chithrananda et al. (2020); Yuksel et al. (2023); Bagal et al. (2021).

Latent-variable molecular models add a continuous representation between discrete molecules and decoded structures. The VAE framework provides an encoder–decoder formulation for such continuous latent spaces Kingma & Welling (2014), and molecular VAEs showed that this geometry can support interpolation and optimization in chemical design Gomez-Bombarelli et al. (2018). However, reconstruction quality or latent smoothness does not imply semantic disentanglement. A model may organize latent space around dataset frequency, sequence length, or molecular size rather than chemically interpretable factors. Our work therefore asks whether an unsupervised Transformer-VAE latent space contains simple directions that remain meaningful after confound-aware analysis and decoded-molecule validation.

2.3 Latent traversal and molecular optimization

Latent optimization methods treat a trained molecular generator as a continuous search space. Constrained Bayesian optimization and latent inceptionism use surrogate or oracle-driven objectives to search VAE latent spaces for improved molecules Griffiths & Hernandez-Lobato (2020); Eckmann et al. (2022). Linear manipulation methods impose a stronger interpretability assumption: a reusable direction in latent space should correspond to a molecular change across many seeds. For example, linear decision boundaries have been used to define molecular manipulation directions through their normal vectors Du et al. (2023). ChemFlow generalizes latent traversal by learning vector fields that transport molecular latent distributions toward property or diversity objectives Wei et al. (2024).

Our objective is simpler and more diagnostic. We do not learn a flow or optimize a supervised property objective during generator training. Instead, we test whether linear probes fitted after unsupervised training reveal globally reusable directions in the frozen latent space. This makes the method efficient

and interpretable, but also imposes a stricter validation burden: a direction must produce meaningful decoded molecular changes, not merely high latent-space predictor scores.

2.4 Probing representations and controlling shortcuts

Probing is a standard method for analyzing what information is accessible in learned representations. Linear probes were introduced as a way to test linearly available information in neural features Alain & Bengio (2016), but probe accuracy alone can be misleading without capacity controls and appropriate baselines Hewitt & Liang (2019); Belinkov (2022). This distinction is central for molecular steering. A linear probe is not used here because it is the strongest predictor, but because its weight vector defines a single global direction. Nonlinear probes can reveal additional recoverable information, but their gradients are local and model-dependent, making them better suited for diagnosing nonlinearity than for claiming global steerability.

3 Methodology

3.1 Dataset Representation and Property Panel

Molecules are represented as SELFIES token sequences derived from the original SMILES strings. Raw SMILES provide compact molecular encodings, but their sequence form depends on traversal conventions and can introduce syntactic variability that is not necessarily chemically meaningful. We therefore use SELFIES as the training representation, since they provide a grammar-constrained molecular language while remaining directly tokenizable for sequence modeling (Krenn et al., 2020).

Each molecule is encoded as $x = (x_1, \ldots, x_T)$, where x_t denotes a SELFIES token. Padding is used only for batching, where T is the maximum length after padding. The Transformer-VAE is trained on these tokenized SELFIES sequences, while all downstream molecular descriptors are computed from the corresponding RDKit molecular graph.

For latent-space analysis, we construct a property matrix $Y \in \mathbb{R}^{N \times P}$, where each row corresponds to a molecule, totaling N molecules, and each column corresponds to an RDKit-computed scalar descriptor, totaling P descriptors. These descriptors cover size, polarity, flexibility, ring topology, lipophilicity, three-dimensionality, topological complexity, synthetic accessibility, and drug-likeness. The full descriptor list, definitions, RDKit implementations, and tier assignments are provided in Appendix A.2. This separation keeps the main analysis focused on latent predictability and steerability, while the appendix documents the chemical interpretation of each property.

3.2 Transformer-VAE Formulation

Given a SELFIES-tokenized molecule $x = (x_1, \ldots, x_T)$, the model encodes the token sequence with a Transformer encoder, $H = \mathrm{Enc}_\theta(x) \in \mathbb{R}^{T \times h}$, where θ denotes the weights and h denotes the hidden size. The self-attention mechanism allows the representation to capture dependencies between distant SELFIES tokens, such as branches, ring closures, and functional substructures.

To obtain a molecular latent representation, the encoded token states are aggregated using learned multi-slot pooling. The slot representations are mapped to Gaussian posterior components and combined into a single global posterior, $q_\theta(z \mid x) = \mathcal{N}\left(\mu, \mathrm{diag}(\sigma^2)\right)$. Here, $\mu \in \mathbb{R}^d$ and $\sigma \in \mathbb{R}^d_{>0}$ denote the mean and standard deviation of the posterior over the d-dimensional latent variable z. Details of the slot construction and confidence-weighted aggregation are provided in Appendix B.1.

The latent variable is sampled using the reparameterization trick, $z = \mu + \sigma \odot \epsilon, \epsilon \sim \mathcal{N}(0, I)$.

A Transformer decoder reconstructs the SELFIES sequence autoregressively from the latent vector. During training, the decoder receives the shifted ground-truth sequence under teacher forcing, while causal masking prevents access to future tokens. Cross-attention to the latent memory routes reconstruction through z, which is later used for property probing, confound analysis and traversal.

The model is trained with the VAE objective: $\mathcal{L} = \mathcal{L}_{\mathrm{rec}} + \beta D_{\mathrm{KL}}\left(q_\theta(z \mid x) \parallel \mathcal{N}(0, I)\right)$,

where $\mathcal{L}_{\mathrm{rec}}$ is the reconstruction loss, and β is the regularization coefficient. The reconstruction term preserves molecular identity, while the KL term regularizes the posterior toward a smooth latent prior.

3.3 VAE Benchmarking Metrics

We evaluate the VAE with five generative checks: reconstruction, novelty, validity, uniqueness, and family retention. Reconstruction measures exact sequence recovery. For a generated structure x, novelty is defined from its maximum Tanimoto similarity to the training set using Morgan fingerprints (radius 2, 2048 bits): a distinct valid sample is counted as novel when its nearest-training similarity is below 0.70, with sensitivity at 0.80 and 0.90 reported in Appendix B.2. Validity is computed by decoding sampled latent vectors to SELFIES, converting them to SMILES, and applying RDKit sanitization. Uniqueness is computed over the valid canonicalized SMILES, so duplicate molecular structures are not counted separately.

Family retention measures whether interpolation paths between molecules from the same functional family preserve that family across decoded steps. Full details are provided in Appendix B.3.

3.4 Confound-Aware Latent Property Analysis

3.4.1 Property and Confound Prediction from Latent Space

For each molecule x_i, the encoder produces a latent representation $z_i = E(x_i) \in \mathbb{R}^d$. Each latent point is associated with two sets of scalar targets: a property panel $Y \in \mathbb{R}^{N\times K}$, containing K molecular descriptors, and a confound panel $C \in \mathbb{R}^{N\times M}$, containing M sequence-level statistics derived from the SELFIES representation.

We include the confound panel because molecular properties can correlate with superficial string-level features, such as sequence length or special-token counts. A probe trained directly on z_i may therefore achieve high predictive performance by exploiting these shortcut variables rather than chemically meaningful latent organization.

To quantify this, we fit one linear probe per target using frozen latent representations: $\hat{y}_i^{(k)} = w_k^\top z_i + b_k$, $\hat{c}_i^{(m)} = u_m^\top z_i + a_m$. Here, y and c denote property and confound targets, w_k and u_m are probe weights, and b_k and a_m are intercepts. Performance is evaluated on held-out data using R^2.

This step is diagnostic. High R^2 for a molecular property indicates that the latent space contains predictive information, but not necessarily that the signal is chemically specific. If the same latent space also predicts confounds that are correlated with the property, the apparent property signal may be shortcut-driven. This motivates the residualization procedure introduced next, where the component of each property explainable from the confound panel is removed before interpreting latent-property directions.

3.4.2 Residualization against SELFIES Confounds

Let Z, Y, and C denote the frozen latent representations, molecular property panel, and SELFIES confound panel. For each property y, we fit a confound-only predictor $\hat{y}^{(C)} = g(C)$, and define the residualized target: $y_{\text{res}} = y - \hat{y}^{(C)}$.

Latent probes are then evaluated on y_{res} rather than on the raw property: $\widehat{y}_{\text{res}} = w_{\text{res}}^\top z + b_{\text{res}}$.

This tests whether latent predictability remains after removing the component explainable from SELFIES-level artifacts. A large performance drop indicates confound-mediated signal, while stable performance suggests that the latent space captures information beyond the measured shortcuts. Residualized targets are therefore used diagnostically, and we additionally repeat the analysis with leakage-controlled, five-fold cross-fitted nonlinear nuisance MLPs (Appendix D.3).

3.5 Linear Probes as Global Property Directions

We use linear probes to test whether a property is organized along a steerable latent direction. Given frozen latent codes $z_i = E(x_i) \in \mathbb{R}^d$, where d is the model dimension, and scalar property values y_i, we fit a linear probe $\hat{y}_i = w^\top z_i + b$. The probe weight w is then interpreted as the global latent direction associated with the property.

For a perturbation $z' = z + \epsilon w$, the predicted property changes as $\hat{y}(z') = \hat{y}(z) + \epsilon \|w\|_2^2$. Thus, moving along w increases the predicted property under the linear approximation, while moving along $-w$ decreases it. Since $\nabla_z \hat{y}(z) = w$, the probe defines a single global direction field.

Successful traversal along w therefore indicates that the property is approximately globally linear in latent space. Failure or instability suggests that the property is weakly encoded, local, or not organized along a single global direction. Additionally, once a property direction has been learned, improving the surrogate objective does not require iterative gradient-based optimization, black-box search, or repeated property-oracle queries in latent space. Further discussion is given in Appendix E.1.

3.6 Nonlinear Probe Formulation and Local Steering

The linear probe defines a single global direction only when a property is approximately affine in the latent space. Properties with weak or moderate linear performance may still be encoded in the latent representation, but arranged along curved or locally varying manifolds. We therefore use nonlinear probes as a diagnostic extension of the linear direction study, focusing on the remaining non-trivial descriptors after excluding strongly size-confounded targets.

For each selected property, we fit a multilayer perceptron on the same frozen latent representations and train/validation/test splits used for the linear probes. Latent vectors are standardized using training-set statistics, $\tilde{z}_i = \frac{z_i - \mu_Z}{\sigma_Z}$, and targets are standardized analogously during optimization.

For a scalar target y, the nonlinear probe is $\hat{y}_i = f_\phi(\tilde{z}_i)$, where f_ϕ is a multi-layer perceptron (MLP), with ReLU nonlinearities. The model is trained by minimizing mean-squared error on the training set, with model selection by validation R^2 and final reporting on the held-out test set.

The same procedure is applied to both raw and residual targets. For the raw target, $\hat{y}_i = f_{\phi,y}(\tilde{z}_i)$, whereas for the confound-corrected target, $\hat{y}_{\mathrm{res},i} = f_{\phi,y_{\mathrm{res}}}(\tilde{z}_i)$. Thus, the raw MLP measures whether additional nonlinear property information is present in the latent code, while the residual MLP tests whether that nonlinear information remains after removing SELFIES-level confounds.

Additional information about the use of MLP nonlinear probing for gradient optimization of properties from the latent can be found in the Appendix F.1.

4 Experiments

4.1 Experimental Setup

We train and evaluate Transformer-VAE models on 794,403 RDKit valid SELFIES molecules. SELFIES strings are tokenized with start and end tokens, and padded to a maximum sequence length of 77. The saved tokenizer contains 110 tokens, including PAD, SOS, and EOS; the autoregressive implementation additionally reserves a MASK token. The dataset is split into train, validation, and test partitions using an 80/10/10 split.

The molecular descriptor panel contains the properties described in Appendix A.2. To control for representation-level artifacts, we also compute a SELFIES confound panel consisting of token length, branch-token count, ring-token count, and token entropy.

4.2 Model Variants and Ablation Settings

We compare three Transformer-VAE variants. The two non-autoregressive variants are architectural controls assembled from established components rather than proposed architectural contributions: Simple Attention combines standard multi-head attention (Vaswani et al., 2017) with learned-query pooling inspired by Set Transformer (Lee et al., 2019), whereas Linear Attention uses the positive feature-map formulation introduced for linear Transformers (Katharopoulos et al., 2020). Non-autoregressive molecular VAEs also predate this work (Kwon et al., 2019). The autoregressive model is used for the main confound-aware analysis. Table 1 summarizes the architectures.

The comparison is designed to separate reconstruction quality from latent-space utility. High reconstruction is necessary for a useful molecular VAE, but it does not by itself show that molecular properties are organized as chemically meaningful latent directions.

Table 1: Transformer-VAE model variants.

Model name	Decoder type	Latent dim.	Hidden size	Encoder blocks	Decoder blocks	Slot pooling	Decoder attention
Linear Attention Non-AR	Non-AR	512	256	1	1	Linear slot pooling	Linear cross-attention
Simple Attention Non-AR	Non-AR	256	256	1	1	Multi-head slot pooling	Standard cross-attention
Autoregressive MultiSlotting	AR	256	256	3	2	Multi-slot pooling	Causal Transformer decoder with latent memory

Table 2: Reconstruction, generation, and interpolation benchmarks. Best available values are bolded. FP novelty is reported only where recomputed with the fingerprint protocol.

Model	Token acc. test/val	Seq. acc. test/val	Gen. validity	Uniqueness	FP novelty	Interp. valid frac.	Family retention
Linear Attention Non-AR	**0.999929**	**0.997671**	0.9226	0.8071	0.9871	**1.0000**	0.8225
Simple Attention Non-AR	0.998709	0.961443	0.8700	0.7700	0.9875	**1.0000**	0.6926
Autoregressive MultiSlotting	0.9951	0.9780	**1.0000**	**0.9966**	**0.9976**	**1.0000**	**0.8512**

4.3 Model Selection and Baseline Latent Quality

Table 2 summarizes reconstruction for test/val, and generation and interpolation results, defined previously in the methodology for N=5000 randomly sampled latents. Linear Attention Non-AR achieves the strongest reconstruction. Simple Attention Non-AR provides a complete earlier baseline for latent-space analysis. Autoregressive MultiSlotting is selected as the main model for property-organization analysis because it gives the strongest raw and residual property-prediction results.

These results support model selection based on latent organization rather than reconstruction alone. For the selected autoregressive model, the fingerprint audit yields 4,983 distinct valid structures from 5,000 samples, with 4,971/4,983 (99.7592%) below maximum training-set Tanimoto 0.70 (Appendix B.2). Autoregressive MultiSlotting also provides the strongest evidence that chemically meaningful descriptors remain predictable after controlling for SELFIES sequence confounds; traversal-based validation follows.

4.4 Confound-Aware Latent Property Prediction

We evaluate whether latent property prediction reflects molecular structure rather than simple SELFIES statistics. The analysis asks: first, how well the latent space predicts RDKit descriptors; second, how strongly it encodes SELFIES confounds; and third, how much property signal remains after residualizing descriptors against these confounds.

Both non-autoregressive models strongly encode SELFIES statistics. This motivates residualized probing. HeavyAtomCount, MolWt, and BertzCT are strongly correlated with SELFIES length, with Spearman correlations of 0.9759, 0.9464, and 0.9042, respectively, as shown in Appendix D.1. Raw probe performance on such descriptors is therefore not sufficient evidence of chemically meaningful latent organization.

Autoregressive MultiSlotting gives the best raw and residual R^2 across the key descriptor panel (Tables 3 and 4). For the main non-size descriptors, its residual performance remains high despite modest confound-to-property predictability. This conclusion survives nonlinear nuisance control: across six traversal-relevant properties, nonlinear-residual latent R^2 remains 0.563–0.761, while fresh nonlinear models recover at most $R^2 = 0.0091$ from the measured confound panel (Appendix D.3).

Appendix Table 15 summarizes the descriptor regimes for the studied properties.

4.5 Linear Traversal Results

To test whether the learned linear probe directions correspond to controllable axes in latent space, we traversed along each regression direction and decoded the resulting latent points back to molecules. For each property, we repeated the experiment for $N = 50$ randomly sampled seed molecules. We summarize the response by the median trajectory together with the interquartile interval across seeds.

Representative results are shown in Fig. 2. The four displayed properties; cLogP, FractionCSP3, TPSA, and HBA, all show clear monotonic increase along the positive traversal direction. FractionCSP3 spans nearly its full range, cLogP increases steadily, TPSA exhibits the strongest response, and HBA also follows a stable upward trend. These results show that the corresponding probe directions are not only predictive, but operational as steering directions after decoding.

Table 3: Raw latent-to-property R^2 for key molecular descriptors. Best values are bolded.

Model	HAC	MolWt	BertzCT	Ring	A.Ring	cLogP	TPSA	HBA	FCSP3	HBD	Spiro	Bridge	QED	NumRotBond	SA	Mean
Linear Attention	0.9562	0.9136	0.9279	0.8549	0.8423	0.6677	0.6557	0.7071	0.7815	0.2171	0.0484	0.0737	0.3678	0.4734	0.4566	0.5963
Simple Attention	0.9622	0.9174	0.9305	0.8491	0.8498	0.6830	0.6994	0.7587	0.7842	0.2589	0.0509	0.0767	0.3465	0.4844	0.4570	0.6072
AR MultiSlotting	**0.9750**	**0.9480**	**0.9512**	**0.9378**	**0.8937**	**0.7967**	**0.8059**	**0.8008**	**0.8951**	**0.4997**	**0.0878**	**0.1522**	**0.4624**	**0.6914**	**0.6412**	**0.7026**

Table 4: Residualized latent-to-property R^2 after removing SELFIES confounds. Best values are bolded.

Model	HAC	MolWt	BertzCT	Ring	A.Ring	cLogP	TPSA	HBA	FCSP3	HBD	Spiro	Bridge	QED	NumRotBond	SA	Mean
Linear Attention	0.3612	0.3799	0.5475	0.5143	0.6067	0.5406	0.4653	0.4735	0.6770	0.1782	0.0318	0.0472	0.3011	0.2065	0.3887	0.3813
Simple Attention	0.3811	0.3109	0.5627	0.5151	0.6518	0.5770	0.5233	0.5404	0.7098	0.2202	0.0354	0.0443	0.2807	0.1992	0.3848	0.3958
AR MultiSlotting	**0.6554**	**0.5918**	**0.7009**	**0.7944**	**0.7559**	**0.6861**	**0.6863**	**0.6759**	**0.8313**	**0.4624**	**0.0728**	**0.1207**	**0.3907**	**0.4752**	**0.6015**	**0.5668**

Across the full traversal study, the clearest globally monotonic behavior was observed for HeavyAtomCount, cLogP, TPSA, HBA, FractionCSP3, and BertzCT. RingCount (including aromatic) was less regular, which is unsurprising given their discrete ring-based nature. SA-score was unstable under traversal: near-seed molecules remain easier to synthesize, while distant decoded molecules become harder. Importantly, some strongly confounded descriptors, notably HeavyAtomCount and BertzCT, still remained meaningfully traversable. This suggests that the confound structure captured by the model is not reducible to simple SELFIES length or token-count effects.

Crucially, monotonicity is not restricted to raw directions. Traversing directions fitted to residualized targets remains strongly monotonic for FractionCSP3, HBA, TPSA, and cLogP (median per-seed Spearman $\rho = 0.912$, 0.929, 0.915, and 0.959), while reducing alignment with the measured confound subspace. Full raw/residual comparisons, including the more confound-sensitive BertzCT and HeavyAtomCount directions, are in Appendix E.2; inter-property geometry is in Appendix E.4.

4.6 Nonlinear Probe Results

4.6.1 Linear Probe versus MLP Performance

We compare the autoregressive model with the simple-attention baseline on the expanded molecular-property set. The linear-attention baseline is excluded because preliminary traversal experiments showed decode collapse and repeated same-molecule outputs, making an MLP-versus-linear comparison uninformative for linear steerability; details are provided in Appendix E.6.

The linear regression results have already shown that the autoregressive latent space is more predictive under linear readout. Here, we ask whether this advantage persists when the readout is allowed to be nonlinear. Using MLP probes, the autoregressive model remains better than the simple-attention baseline on every expanded property, both before and after confound control.

However, the gap between models decreases with the nonlinear readout. On raw targets, the mean R^2 gap decreases from 0.096 with linear probes (0.703 vs. 0.607) to 0.064 with MLP probes (0.895 vs. 0.831). After residualization, the gap decreases from 0.171 with linear probes (0.567 vs. 0.396) to 0.100 with MLP probes (0.859 vs. 0.759). Full per-property MLP results are reported in Appendix Table 19.

This pattern suggests that both latent spaces contain nonlinear local information about the descriptors, but that the autoregressive model organizes this information in a more globally linear form. This distinction is important because property-directed traversal depends on the existence of stable linear directions, not merely on nonlinear decodability.

4.6.2 Properties with Strong Nonlinear Structure

The MLP comparison shows that nonlinear readouts recover additional descriptor information from the autoregressive latent space. However, this does not imply that the corresponding properties are suitable for linear traversal. A property may be decodable from z while still lacking a single global direction along which it changes monotonically.

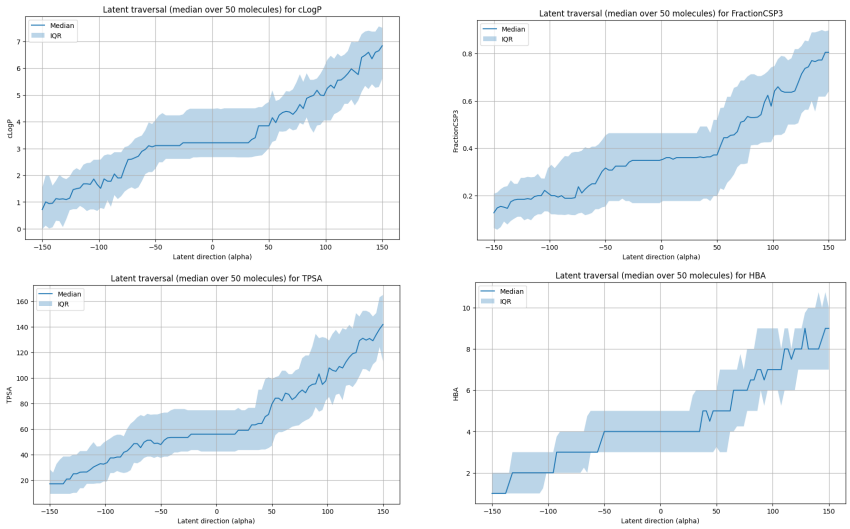

Figure 2: Latent traversals for cLogP, FractionCSP3, TPSA, and HBA (median and interquartile interval over $N = 50$ seed molecules).

We therefore focus on descriptors for which the autoregressive model shows a large gap between linear and MLP probe performance. These cases indicate latent signal that is present, but organized in a nonlinear or interaction-dependent manner. Table 5 reports the clearest examples.

Table 5: Autoregressive descriptors with strong nonlinear structure, indicated by large gaps between linear and MLP probe performance.

Probe / property	HBD	NumRotatableBonds	NumSpiroAtoms	NumBridgeheadAtoms	QED
AR Linear Regression raw	0.500	0.691	0.088	0.152	0.462
AR MLP raw	0.792	0.909	0.628	0.785	0.803
AR Linear Regression resid	0.462	0.475	0.073	0.121	0.391
AR MLP resid	0.778	0.849	0.608	0.771	0.771

NumSpiroAtoms and NumBridgeheadAtoms are the strongest examples: linear probes are nearly ineffective, whereas MLP probes recover substantial predictive power in both the raw and residual settings. HBD, QED and NumRotatableBonds show the same pattern more moderately. These results suggest that such descriptors are represented in the latent space, but through nonlinear combinations of latent features rather than through a stable global axis. Additional insight about ΔR^2 and full per-property results are reported in Appendix F.1.1.

5 Conclusion, Limitations and Future Directions

We presented a confound-aware framework for evaluating whether an unsupervised SELFIES-based Transformer-VAE learns chemically meaningful, steerable latent structure. The autoregressive MultiSlotting model organizes several RDKit descriptors into predictive, operationally steerable directions, while nonlinear probes show that other descriptors remain latently encoded without a single global linear direction. The study is limited to computed RDKit descriptors and therefore does not establish performance on experimental biochemical, pharmacokinetic, toxicity, docking, or synthesis outcomes. In addition, the confound panel captures only a minimal set of SELFIES-level shortcuts, and the steering results are evaluated for one model family and one large SELFIES dataset. Broader validation across datasets, architectures, molecular domains, and decoding strategies remains necessary. Future work should extend confound control to graph-size, scaffold, functional-group, and dataset-frequency effects; combine linear steering with chemical trust-region constraints such as fingerprint similarity, scaffold retention, synthetic accessibility, and multi-objective trade-offs; and explore local nonlinear steering methods based on MLP gradients or learned latent vector fields.

AI USE STATEMENT

Generative AI tools were used for language polishing, grammar review, and formatting assistance. They were not used to generate the scientific ideas, analyses, results, or conclusions. All final content was reviewed and approved by the authors.

REPRODUCIBILITY STATEMENT

All implementation details, hyperparameters, dataset preprocessing procedures, evaluation protocols, and additional experimental results required to reproduce the study are provided in the main text and appendices (mainly Appendix A). Code and reproducibility artifacts are available in the anonymous repository linked in the paper.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Marvin Alberts, Oliver Schilter, Federico Zipoli, Nina Hartrampf, and Teodoro Laino. Unraveling molecular structure: A multimodal spectroscopic dataset for chemistry. *Advances in Neural Information Processing Systems*, 37:125780–125808, 2024.

Viraj Bagal, Rishal Aggarwal, P. K. Vinod, and U. Deva Priyakumar. MolGPT: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 62(9): 2064–2076, 2021. doi: 10.1021/acs.jcim.1c00600.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Esben Jannik Bjerrum. SMILES enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076*, 2017.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. ChemBERTa: Large-scale self-supervised pretraining for molecular property prediction. *arXiv preprint arXiv:2010.09885*, 2020. doi: 10.48550/arXiv.2010.09885.

Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1423.

Yuanqi Du, Tianfan Fu, Jimeng Sun, and Shengchao Liu. MolGenSurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500*, 2022.

Yuanqi Du, Xinhao Liu, Neil M. Shah, Shengchao Liu, Jie Zhang, and Bolei Zhou. ChemSpace: Interpretable and interactive chemical space exploration. *Transactions on Machine Learning Research*, 2023.

Peter Eckmann, Kunyang Sun, Bo Zhao, Mudong Feng, Michael K. Gilson, and Rose Yu. LIMO: Latent inceptionism for targeted molecule generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5777–5792. PMLR, 2022.

Tianfan Fu, Wenhao Gao, Connor W. Coley, and Jimeng Sun. Reinforced genetic algorithm for structure-based drug design. In *Advances in Neural Information Processing Systems*, volume 35, pages 12325–12338, 2022.

Rafael Gomez-Bombarelli, Jennifer N. Wei, David Duvenaud, Jose Miguel Hernandez-Lobato, Benjamin Sanchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alan Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572.

Ryan-Rhys Griffiths and Jose Miguel Hernandez-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 11(2):577–586, 2020. doi: 10.1039/C9SC04026A.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Shion Honda, Shoi Shi, and Hiroki R. Ueda. SMILES transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738*, 2019. doi: 10.48550/arXiv.1911.04738.

Jan H. Jensen. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332. PMLR, 2018.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR, 2020.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Mario Krenn, Florian Hase, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020. doi: 10.1088/2632-2153/aba947.

Youngchun Kwon, Jiho Yoo, Youn-Suk Choi, Won-Joon Son, Dongseon Lee, and Seokho Kang. Efficient learning of non-autoregressive graph variational autoencoders for molecular graph generation. *Journal of Cheminformatics*, 11:70, 2019. doi: 10.1186/s13321-019-0396-x.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 2019.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118.

Benjamin Sanchez-Lengeling and Alan Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018. doi: 10.1126/science.aat2663.

Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Pankaj Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature Reviews Drug Discovery*, 18(6):463–477, 2019. doi: 10.1038/s41573-019-0024-5.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

Guanghao Wei, Yining Huang, Chenru Duan, Yue Song, and Yuanqi Du. Navigating chemical space with latent flows. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988. doi: 10.1021/ci00057a005.

Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Atakan Yuksel, Erva Ulusoy, Atabey Unlu, and Tunca Dogan. SELFormer: Molecular representation learning via SELFIES language models. *arXiv preprint arXiv:2304.04662*, 2023. doi: 10.48550/arXiv.2304.04662.

Chengxi Zang and Fei Wang. MoFlow: An invertible flow model for generating molecular graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 617–626, 2020. doi: 10.1145/3394486.3403104.

Zhou Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific Reports*, 9:10752, 2019. doi: 10.1038/s41598-019-47148-x.

APPENDIX

A CODE, DATASET, AND ADDITIONAL DATA INFO

A.1 EXPERIMENTAL SETUP AND REPRODUCIBILITY

All code, notebooks, model definitions, checkpoints, and curated artifacts are available in the anonymized repository: `https://anonymous.4open.science/r/MolecuLA-7A68/`. All paths are repository-relative. The released checkpoints are treated as fixed pretrained inputs, while downstream artifacts can be regenerated from `data/`, `checkpoints/`, `models/`, and `study/notebooks/`. The large dataset CSV and checkpoints are stored as Git LFS assets; after cloning, `git lfs pull` should be run before executing notebooks.

Randomness and determinism. Unless otherwise specified, all experiments use `SEED = 42`. Dataset splits use `train_test_split(..., random_state=42, shuffle=True)` for both split stages. The notebooks seed Python `random`, NumPy, `torch.manual_seed`, and `torch.cuda.manual_seed_all` when CUDA is available; explicit generators use `np.random.default_rng(42)` or documented offsets from this seed.

Exact floating-point values may vary slightly across PyTorch, CUDA, cuDNN, BLAS, GPU model, and RDKit versions. The intended reproducibility contract is therefore fixed dataset rows, tokenizer, splits, checkpoint weights, model settings, probe grids, random seeds, and output schemas, with small numerical differences expected across machines.

Dataset, tokenization, and splits. The molecular dataset is derived from the dataset introduced by Alberts et al. (2024). The original dataset contains molecular structures together with multiple experimental and predicted molecular-property modalities, including NMR- and mass-spectrometry-related annotations. In this study, we use only the molecular structure information: the SMILES column is extracted and converted to SELFIES for model training and evaluation. The processed SMILES/SELFIES dataset used in this work is included in the anonymized repository and is sufficient to reproduce all experiments reported here.

The processed dataset is loaded from `data/smiles_selfies_full.csv` and must contain at least `smiles` and `selfies` columns. The released tokenizer is `data/selfies_tokenizer.json`, with `max_len = 77` and `vocab_size = 110` for the non-autoregressive checkpoints. SELFIES strings are encoded by prepending `<SOS>`, appending `<EOS>`, and padding with `<PAD>`.

The autoregressive notebook constructs its own vocabulary with `<PAD>`, `<SOS>`, `<EOS>`, and `MASK`; the corresponding checkpoint records `vocab_size = 111`. This difference is intentional and is handled by the autoregressive model path.

The standard split is 80% train, 10% validation, and 10% test. For the released dataset this gives 635,522 training rows, 79,440 validation rows, and 79,441 test rows. RDKit-derived panels retain invalid molecules in the table but mark them with `is_rdkit_valid`; probe fitting and property summaries use valid molecules with finite values for the relevant target.

Model settings. The model registry in `models/registry.py` is the source of truth for the public model configurations.

Table 6: Transformer-VAE model settings used in the released checkpoints.

Model	Hidden size	Latent size	Max length	Heads	Slots	Encoder/decoder depth	Checkpoint vocab	Checkpoint epoch
`linear_attention`	256	512	77	8	8	1 shared non-AR layer	110	50
`simple_attention`	256	256	77	8	8	1 shared non-AR layer	110	50
`autoregressive`	256	256	77	8	8	3 encoder layers, 2 decoder layers	111	50

The checkpoints contain model weights, optimizer state, training history, checkpoint epoch, and vocabulary size. The optimizer is AdamW with learning rate `3e-4`, betas `(0.9, 0.999)`, epsilon `1e-8`, and weight decay `1e-2`. All three architectures use Transformer-style attention blocks with

dropout $p = 0.1$ inside the feed-forward blocks. The non-autoregressive blocks use feed-forward width $4\times$ hidden size, while the autoregressive blocks use $2\times$ hidden size.

Table 7: Loss terms used by the three Transformer-VAE variants.

Model	Reconstruction term	KL term	Length term
`linear_attention`	token cross entropy over non-pad targets	default $\beta = 0.01$	MSE on predicted length, $\alpha = 0.1$
`simple_attention`	token cross entropy over non-pad targets	default $\beta = 0.01$	MSE on predicted length, $\alpha = 1.0$
`autoregressive`	token cross entropy with `ignore_index=<PAD>`	default $\beta = 0.01$, overridden by schedule	none

Training. The notebook `study/notebooks/00_data_tokenization_and_training.ipynb` contains the rerunnable training workflow. Training was performed on a single NVIDIA A100 40GB GPU and took approximately 2 hours 30 minutes for 50 epochs. The released checkpoints can be used directly without retraining, but rerunning the notebook with the same settings regenerates the models from the same tokenization and split logic.

Table 8: Autoregressive training settings.

Setting	Value
epochs	50
batch size	512
hidden size	256
latent size	256
attention heads	8
number of slots	8
encoder layers	3
decoder layers	2
optimizer	AdamW
learning rate	`3e-4`
betas	`(0.9, 0.999)`
epsilon	`1e-8`
weight decay	`1e-2`
initial `beta` variable	`0.01`
effective beta schedule	`(epoch % cycle_length / cycle_length) * max_beta`
`max_beta`	`0.03`
`cycle_length`	15
corruption	`False`
training loader shuffle	`True`
validation loader shuffle	`False`
notebook `num_workers`	1
checkpoint selection	best validation sequence accuracy

Latent encoding and model-quality benchmarks. Latents are encoded with the model in evaluation mode and without gradient tracking. The helper in `study/common/latents.py` returns the encoder mean μ as `float32`; its default batch size is 256. The main benchmark notebooks use `SEED = 42`, batch size 1024 for reconstruction, prior-sampling, and interpolation evaluations, and `N_STEPS = 11` for interpolation tests. Family-retention tests use seed 42 and a minimum family group size of 20.

Quality outputs include token accuracy, sequence accuracy, reconstruction validity, prior validity, interpolation validity, novelty and uniqueness summaries where available, and family-retention summaries. The curated result folders contain compact CSV/JSON summaries and figures only; regenerated latent arrays and full decode caches are excluded.

Linear probes, residualization, and Ridge directions. The notebook `study/notebooks/02_linear_probes_panels_and_residuals.ipynb` is the canonical workflow for the linear probe analysis. The main reported latent-to-property results, especially for the two ablation transformers, use ordinary linear regression to measure how much property signal is available through a single global linear readout. For each target, finite rows are selected, latent coordinates are standardized using train rows, target values are standardized using train rows, and train/validation/test R^2 scores are reported.

Ridge models are used in auxiliary places where regularization is useful, especially for residualization, confound prediction, and some direction-fitting utilities. Residualization fits a multivariate $C \to Y$ Ridge model on train rows, predicts all valid rows, and stores `resid_{property}` columns. The residual probe then repeats the same linear-probe procedure on these residual targets.

Table 9: Linear-probe and residualization settings.

Setting	Value
`SEED`	42
latent encoding `BATCH_SIZE`	1024
split random state	42
input scaling	`StandardScaler` fit on train rows
target scaling	`StandardScaler` fit on train rows
main reported probe	ordinary linear regression
reported metrics	train/validation/test R^2
Ridge alpha grid, where used	`np.logspace(-3, 3, 13)`
residualization model	multivariate Ridge, $C \to Y$

Autoregressive Step 3 source notebooks used fixed Ridge settings for some compact recovered tables and direction utilities: `alpha=1e-2` for $Z \to Y$ and $Z \to Y_{\text{residual}}$, `alpha=1e1` for $C \to Y$ residualization, and `alpha=1.0` for $Z \to C$ confound directions.

MLP probe settings. The notebook `study/notebooks/04_mlp_probes_all_properties.ipynb` is the canonical MLP-probe workflow. It compares nonlinear MLP probes against the linear baselines on raw and residualized targets.

Table 10: Canonical MLP probe settings.

Setting	Value
`SEED`	42
`BATCH_SIZE`	8192
`MAX_EPOCHS`	15
`MIN_EPOCHS`	6
`PATIENCE`	4
minimum validation improvement	10^{-3} in validation R^2
learning rate	10^{-3}
weight decay	10^{-4}
hidden width	256
architecture	Linear, ReLU, Linear, ReLU, Linear
optimizer	AdamW
loss	MSE on standardized target
input scaling	latent train mean/std; std below 10^{-8} replaced by 1.0
target scaling	train target mean/std; std below 10^{-8} replaced by 1.0
AMP	enabled when CUDA is available

The MLP notebook stores the best in-memory state by validation R^2. Training stops after at least `MIN_EPOCHS` once `PATIENCE` consecutive epochs fail to improve validation R^2 by more than 10^{-3}. Reported scores are computed after inverse-transforming predictions back to original target units.

The autoregressive source MLP notebooks used the same batch size, maximum epochs, patience, learning rate, weight decay, and hidden width, but the compact recovered MLP tables come from an architecture `Linear -> GELU -> LayerNorm -> Linear -> GELU -> Linear`. These source notebooks early-stopped on validation-loss improvement greater than 10^{-5}, with patience 4.

Latent traversal and direction settings. The traversal analysis uses raw-property directions fitted with ordinary linear regression. For each target property, latent means z and property values y are split with `test_size=0.2` and `random_state=42`. A `StandardScaler`+`LinearRegression` pipeline is fitted on the training split. The learned coefficient vector is converted back to the original latent coordinate scale by dividing by the scaler standard deviation, then normalized to unit norm before traversal.

Two traversal protocols are used. The dense single-origin traversal is used for cLogP and starts from the latent origin $z_0 = 0$, traversing

$$z(\alpha) = z_0 + \alpha\hat{w}, \qquad \alpha \in [-200, 200],$$

with 5000 evenly spaced steps. The multi-seed traversal is used for the main property panels. It samples 50 dataset latents as seeds and traverses each seed along the fitted raw-property direction with 100 evenly spaced steps over

$$\alpha \in [-150, 150].$$

Decoded molecules are scored with RDKit, and the reported traversal curves show the median property value across the 50 seeds, with the interquartile range shown as a band. This protocol is applied to F.CSP3, TPSA, cLogP, HBA, MolWt, BertzCT, HAC, Ring, and A.Ring.

Table 11: Latent traversal settings used in the source traversal notebook.

Setting	Value
direction model	`StandardScaler` + ordinary `LinearRegression`
direction target	raw property value
train/test split	`test_size=0.2`, `random_state=42`
direction rescaling	divide coefficient vector by latent scaler std
direction normalization	unit ℓ_2 norm
dense traversal seed	latent origin, $z_0 = 0$
dense traversal property	cLogP
dense traversal alphas	`np.linspace(-200.0, 200.0, 5000)`
multi-seed traversal seeds	50 dataset latents
multi-seed traversal alphas	`np.linspace(-150.0, 150.0, 100)`
multi-seed traversal properties	F.CSP3, TPSA, cLogP, HBA, MolWt, BertzCT, HAC, Ring, A.Ring
decode batch size	4096 for multi-seed traversal decoding
reported traversal summary	median with interquartile range across seeds

Output regeneration policy. The notebooks regenerate large panels, latent matrices, direction arrays, MLP weights, and decode caches locally. The repository commits only compact artifacts needed for inspection: summary CSV files, JSON manifests, selected figures, checkpoints, and data assets tracked by Git LFS.

Curated evidence is stored under `study/results/`. This includes benchmark and family-retention metrics, confound summaries, compact R^2/correlation tables, direction-quality tables, single-property traversal summaries, representative figures, MLP-vs-linear probe summaries, and ablation comparison summaries. Large regenerated assets such as full panels, latent arrays, `.npz` direction caches, duplicate large CSV files, and MLP weight files are excluded.

A.2 Property Panel and Descriptor Tiers

This appendix details the molecular property panel used for latent-space probing listed in Table 12. All properties are computed from RDKit molecular graphs after decoding or standardizing the corresponding molecules; they are not used as supervision during VAE training. We organize the descriptors into three tiers according to interpretability and expected sensitivity to representation-level confounds.

Tier 1: simple size and count descriptors. Tier 1 contains descriptors that are chemically interpretable but often strongly correlated with coarse SELFIES statistics such as sequence length, token count, or molecular size. These properties are useful sanity checks for whether the latent space encodes basic molecular structure, but high predictability may partly reflect representation-level scale effects.

Tier 2: physicochemical and structural descriptors. Tier 2 contains descriptors that remain graph-computable but encode more specific chemical information. These properties probe polarity, lipophilicity, saturation, flexibility, rigidity, and local three-dimensional structure. They are more informative for latent direction discovery because they are less reducible to a single notion of molecular size.

Tier 3: composite and graph-complexity descriptors. Tier 3 contains higher-level descriptors that combine multiple structural or physicochemical factors. These targets are more demanding probes of the latent space because they summarize non-local graph organization or aggregate several simpler molecular properties into a single scalar score.

Table 12: Molecular descriptors used in the property panel, grouped according to the tier system used in the analysis.

Tier	Descriptor	Chemical interpretation
1	MolWt	Molecular size and mass
1	HeavyAtomCount	Number of non-hydrogen atoms
1	HBD	Hydrogen-bond donor count
1	HBA	Hydrogen-bond acceptor count
1	RingCount	Number of rings
1	AromaticRingCount	Number of aromatic rings
1	NumRotatableBonds	Molecular flexibility
2	cLogP	Crippen lipophilicity estimate
2	TPSA	Topological polar surface area
2	FractionCSP3	Carbon saturation and three-dimensionality
2	NumSpiroAtoms	Spiro-center count and rigid 3D ring junctions
2	NumBridgeheadAtoms	Ring fusion complexity and rigidity
3	BertzCT	Graph-topological complexity
3	SA score	Estimated synthetic accessibility
3	QED	Quantitative estimate of drug-likeness

Tier 1 descriptors primarily test whether the latent representation captures coarse molecular scale and simple count information. Tier 2 descriptors probe more chemically specific axes, such as polarity, lipophilicity, saturation, and ring-system rigidity. Tier 3 descriptors are the most composite: BertzCT summarizes graph complexity, SA score combines fragment rarity with structural penalties, and QED aggregates several physicochemical desirability terms. This tiering is used only for interpretation; all descriptors are treated uniformly as scalar targets in the probing experiments.

B VAE ARCHITECTURE AND BENCHMARKING

B.1 TRANSFORMER-VAE ARCHITECTURE DETAILS

The Transformer-VAE operates on SELFIES-tokenized molecules $x = (x_1, \ldots, x_T)$, where T denotes the sequence length and x_t is the token at position t. Token embeddings are combined with positional encodings and processed by Transformer encoder blocks, producing contextual token representations

$$H = \mathrm{Enc}_\theta(x) \in \mathbb{R}^{T \times h},$$

where Enc_θ is the Transformer encoder with parameters θ, h is the hidden dimension, and $H = (h_1, \ldots, h_T)$ contains the encoded token states. Multi-head self-attention enables each non-padding token to attend to all other tokens, allowing the encoder to capture non-local molecular dependencies such as ring closures, branches, and functional groups that may be separated in the SELFIES string.

Rather than compressing the sequence through a single pooled vector, the encoder uses learned multi-slot pooling. A set of learned slot queries $\{q_k\}_{k=1}^{K_s}$, where K_s is the number of slots, attends over the encoded token states:

$$s_k = \sum_{t=1}^{T} \alpha_{kt} V h_t, \qquad \alpha_{kt} = \mathrm{softmax}_t \left(q_k^\top K h_t \right),$$

where s_k denotes the k-th latent slot, q_k is its learned query vector, K and V are learned key and value projection matrices, and α_{kt} is the attention weight assigned by slot k to token position t. The softmax is taken over the token index t. This allows different slots to specialize to different aspects of the molecular representation, which is useful when molecular properties are distributed across multiple structural motifs.

Each slot is mapped to a Gaussian posterior component:

$$\mu_k = W_\mu s_k, \qquad \log \sigma_k^2 = W_\sigma s_k,$$

where μ_k and σ_k^2 are the mean and diagonal variance of the k-th slot-level Gaussian component, and W_μ and W_σ are learned linear projections.

The slot-level Gaussians are combined into a global posterior using confidence-weighted latent pooling. Slots with lower predicted uncertainty receive larger weights:

$$\alpha_k = \mathrm{softmax}_k \left(\frac{q^\top K \mu_k}{\tau} + \lambda c_k \right), \qquad c_k = -\log \sum_j \exp\left(\log \sigma_{k,j}^2\right).$$

Here, α_k is the normalized weight assigned to slot k, q is a learned global pooling query, K is a learned projection matrix applied to the slot mean, $\tau > 0$ is a temperature parameter, λ controls the influence of the confidence term, and c_k is a scalar confidence score for slot k. The index j runs over latent dimensions, and $\sigma_{k,j}^2$ is the variance of the j-th latent dimension for slot k. Since c_k decreases with the total predicted variance, slots with lower uncertainty receive larger confidence scores.

The final posterior parameters are

$$\mu = \sum_k \alpha_k \mu_k, \qquad \sigma^2 = \sum_k \alpha_k \sigma_k^2,$$

where μ and σ^2 denote the mean and diagonal variance of the global approximate posterior. This yields

$$q_\theta(z \mid x) = \mathcal{N}\left(\mu, \mathrm{diag}(\sigma^2)\right),$$

where z is the latent molecular representation and $q_\theta(z \mid x)$ is the approximate posterior distribution parameterized by the encoder.

The latent vector is sampled using the reparameterization trick:

$$z = \mu + \sigma \odot \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I),$$

where ϵ is standard Gaussian noise, I is the identity matrix, σ is the elementwise standard deviation, and $\odot$ denotes elementwise multiplication.

The decoder is an autoregressive Transformer decoder. The latent vector is projected into a memory representation,

$$m = W_m z,$$

where W_m is a learned projection matrix and m is the latent memory used by the decoder. The decoder reconstructs the SELFIES sequence as

$$p_\theta(x \mid z) = \prod_{t=1}^{T} p_\theta\left(x_t \mid x_{<t}, z\right),$$

where $x_{<t}$ denotes the previously generated tokens before position t, and $p_\theta(x \mid z)$ is the decoder likelihood parameterized by θ. Training uses teacher forcing with shifted ground-truth inputs. Causal self-attention prevents access to future tokens, while cross-attention allows each decoding position to condition on the latent memory m. This prevents direct encoder–decoder token copying and ensures that reconstruction depends on the global latent representation.

B.2 Fingerprint-based novelty audit

For the selected autoregressive model, we re-audited 5,000 prior samples against the training partition only (635,522 molecules). All 5,000 decoded samples were RDKit-valid, yielding 4,983 distinct canonical structures; the remaining 17 were duplicates rather than invalid generations. We canonicalized structures with RDKit and computed Morgan fingerprints with radius 2 and 2,048 bits; nearest-training similarities were evaluated with FPSim2 (v0.7.4; RDKit v2025.9.3). For each generated structure x, define

$$s(x) = \max_{t \in \mathcal{D}_{\mathrm{train}}} \mathrm{Tanimoto}(\mathrm{fp}(x), \mathrm{fp}(t)).$$

A structure is counted as novel at threshold τ when $s(x) < \tau$. At $\tau = 0.70$, 4,971/4,983 structures (99.7592%) are novel; sensitivity analyses give 99.8395% at 0.80 and 99.8595% at 0.90. The median nearest-training similarity is 0.2373, with 95th and 99th percentiles of 0.3997 and 0.5517. We use 0.70 as the primary operating threshold for this audit, not as a universal definition of chemical novelty.

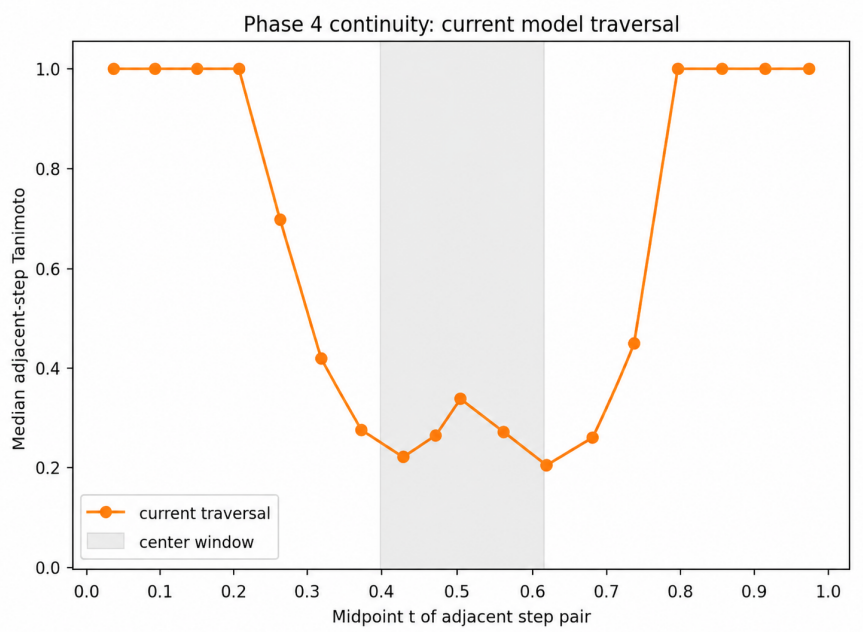

Figure 3: Interpolation continuity for the current model. The curve reports the median adjacent-step Tanimoto similarity over $N = 200$ interpolation paths. The shaded region marks the central interpolation window, where structural transitions are largest.

B.3 Interpolation Smoothness and Family-Conditioned Retention

We evaluate latent-space interpolation as an additional diagnostic of the learned molecular geometry. Given two validation molecules x_1 and x_2 with latent codes

$$z_1 = E(x_1), \qquad z_2 = E(x_2),$$

we construct the linear latent path

$$z(t) = (1-t)z_1 + tz_2, \qquad t \in [0,1].$$

Each intermediate point is decoded and mapped back to a molecular structure. Unless otherwise stated, interpolation paths use $K = 11$ decoded steps and statistics are averaged over $N = 200$ sampled endpoint pairs.

We report two interpolation diagnostics. First, we measure decoded validity at each interpolation step. The model obtains a valid fraction of 1.0 across all interpolation positions, indicating that the decoder remains chemically valid along linear paths between encoded validation molecules. Second, we measure local structural continuity using adjacent-step Tanimoto similarity:

$$s_i = \text{Tanimoto}\left(\text{FP}(\hat{x}(t_i)), \text{FP}(\hat{x}(t_{i+1}))\right),$$

where $\hat{x}(t_i)$ is the decoded molecule at interpolation step t_i and $\text{FP}(\cdot)$ denotes the molecular fingerprint. Figure 3 reports the median adjacent-step Tanimoto similarity as a function of the midpoint between adjacent interpolation steps. The model is most continuous near the endpoints, while the central region exhibits lower similarity, indicating larger structural transitions near the midpoint despite preserved validity.

We further evaluate whether interpolation preserves local semantic structure by performing *family-conditioned interpolation*. In this setting, both endpoints are sampled from the same functional family f, defined by deterministic SMARTS matching on the validation set. For each decoded interpolation point, we compute the retention indicator

$$r_f(t_i) = \mathbf{1}\left[\hat{x}(t_i) \in f\right],$$

Table 13: Functional-family definitions used for family-conditioned interpolation. Families are assigned by SMARTS matching on decoded molecules.

Family	Chemical motif	SMARTS pattern
Alcohol	$R{-}OH$	`[OX2H][CX4]`
Phenol	$Ar{-}OH$	`[OX2H][c]`
Ether	$R{-}O{-}R'$	`[OD2]([#6;!$(C=O)])[#6;!$(C=O)]`
Amine	primary/secondary/tertiary amine	`[NX3;H2,H1,H0;!$(N[C,S,P]=O)]`
Amide	$R{-}C(=O){-}NR_2$	`[NX3][CX3](=[OX1])[#6,#7,#8]`
Carboxylic acid	$R{-}C(=O)OH$	`[CX3](=O)[OX2H1]`
Ester	$R{-}C(=O)OR'$	`[CX3](=O)[OX2H0][#6]`
Aldehyde	$R{-}CHO$	`[CX3H1](=O)[#6]`
Ketone	$R{-}C(=O){-}R'$	`[#6][CX3](=O)[#6]`
Nitrile	$R{-}C{\equiv}N$	`[CX2]#N`
Sulfonamide	$R{-}SO_2{-}NR_2$	`[SX4](=[OX1])(=[OX1])([#6])[NX3]`

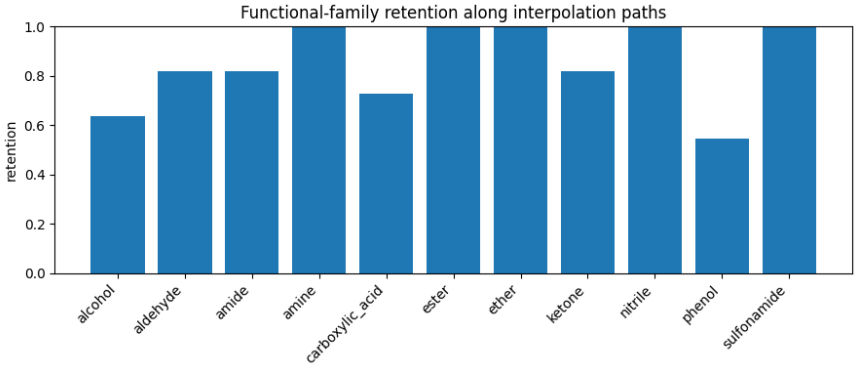

Figure 4: Functional-family retention along family-conditioned interpolation paths. Bars report the average fraction of decoded interpolation steps that remain within the endpoint functional family. All paths have valid fraction 1.0; the variation shown here therefore reflects semantic motif retention rather than decoding validity.

and summarize each family by

$$R_f = \frac{1}{K}\sum_{i=1}^{K} r_f(t_i).$$

Thus, R_f measures the fraction of decoded interpolation steps that remain within the target functional family. This diagnostic separates chemical validity from motif-level semantic continuity: a path may remain valid while losing the functional group shared by its endpoints.

The functional groups used in this analysis are listed in Table 13. They cover common local motifs including alcohols, phenols, ethers, amines, amides, carboxylic acids, esters, aldehydes, ketones, nitriles, and sulfonamides.

Figure 4 reports the resulting family-retention scores. All family-conditioned paths retain perfect decoded validity. Motif retention, however, varies by functional group. Amines, esters, ethers, nitriles, and sulfonamides are retained across all decoded interpolation steps. Aldehydes, amides, and ketones also show high retention. Alcohols, carboxylic acids, and phenols are less stable, with phenols showing the lowest retention. This suggests that some functional motifs correspond to more locally connected regions in latent space, whereas others are more easily disrupted by interpolation.

C Robustness analysis of the latent space

Because held-out R^2 alone does not distinguish genuine latent signal from statistical artifact, we complement probe performance with a set of robustness controls. The goal is to test whether predictive performance is reproducible, whether it collapses under label destruction, and whether the inferred directions depend only on the latent information itself rather than on an arbitrary coordinate system.

C.1 Robustness Tests conducted

First, we assess *bootstrap stability*. For each property, let w denote the probe direction fit on the full training set, and let $w^{(b)}$ denote the direction obtained from bootstrap resample $b \in \{1, \dots, B\}$. Stability is summarized by the sign-aligned cosine similarity $s^{(b)} = \frac{\langle w^{(b)}, w \rangle}{\|w^{(b)}\|_2 \, \|w\|_2}$., $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product.

High agreement indicates that the recovered direction is not a fragile consequence of a particular sample realization, but a reproducible feature of the latent representation.

Second, we apply a *permutation control*. The latent codes are kept fixed, but the property values are randomly permuted within the training split, $y_i^{\pi} = y_{\pi(i)}$, before refitting the probe. If the original R^2 reflects genuine latent–property association, predictive performance should collapse under permutation on validation and test data. This control directly checks that the measured signal is not produced by chance fitting capacity or by regularization artifacts.

Finally, we test *rotation and basis invariance*. Linear probe directions should represent geometric structure in latent space rather than dependence on a particular coordinate basis. Concretely, if the latent codes are transformed by an orthogonal map Q, so that $z_i' = Qz_i, Q^\top Q = I$, then an equivalent predictor exists in the rotated basis:

$$\hat{y}_i = w^\top z_i + b = (Qw)^\top z_i' + b,$$

and held-out predictions should remain unchanged. This verifies that the interpretation is attached to the latent subspace geometry, not to individual coordinates.

A property is therefore treated as robust only when its predictive signal is reproducible across resamples, disappears under permutation, and remains consistent under admissible changes of latent parameterization.

C.2 Bootstrap stability of learned directions

Figure 5 summarizes the bootstrap stability analysis. For each property, the probe was refit across multiple bootstrap resamples of the training set (B = 100) , and the learned directions were compared to the reference direction using cosine similarity. The plotted quantity is therefore a directional stability measure rather than a direct performance metric.

For both raw and residual targets, the median bootstrap cosine is close to 1 for nearly all properties. This indicates that the learned directions are highly reproducible under resampling and are not driven by a small subset of training examples or by instability in the fitting procedure. Importantly, this stability remains high after residualization, showing that the residual-property directions are still well defined in latent space. However, bootstrap stability alone should not be interpreted as evidence that a direction is free of confound influence; rather, it shows that the recovered direction is robust.

C.3 Permutation and random-direction controls

Figure 6 reports two control analyses. The left panel shows the permutation control, in which training labels are randomly permuted before fitting the probe. Under permutation, test-set R^2 values collapse to approximately zero, with some slightly negative values, as expected when the true association between latent representations and labels has been destroyed. This indicates that the non-zero R^2 values observed with the true labels do not arise from trivial artifacts of the training procedure or data split.

The right panel compares the observed alignment between property directions and confound directions to a null distribution obtained from random latent directions. For each random direction, the maximum

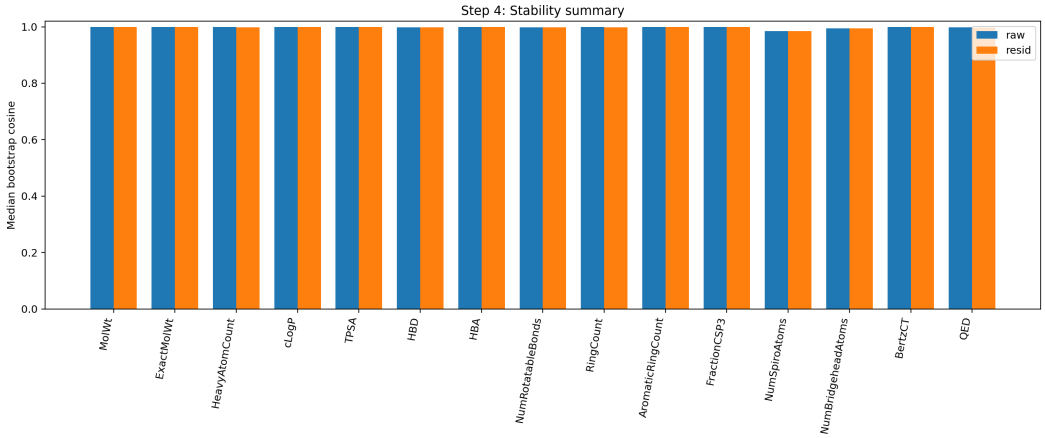

Figure 5: Bootstrap stability of probe directions. Bars report the median cosine similarity between the reference direction and the directions obtained by refitting on bootstrap resamples of the training data, for raw and residual targets.

absolute cosine similarity to the confound directions is computed. The random baseline remains concentrated at relatively low values. The median raw-property alignment lies far above this null distribution, showing that raw property directions are substantially more aligned with confound directions than would be expected by chance. Residualization reduces this alignment, but the median residual-property alignment remains above the random baseline. Thus, residualization suppresses confound-driven signal, but does not render the learned directions fully orthogonal to the confound subspace.

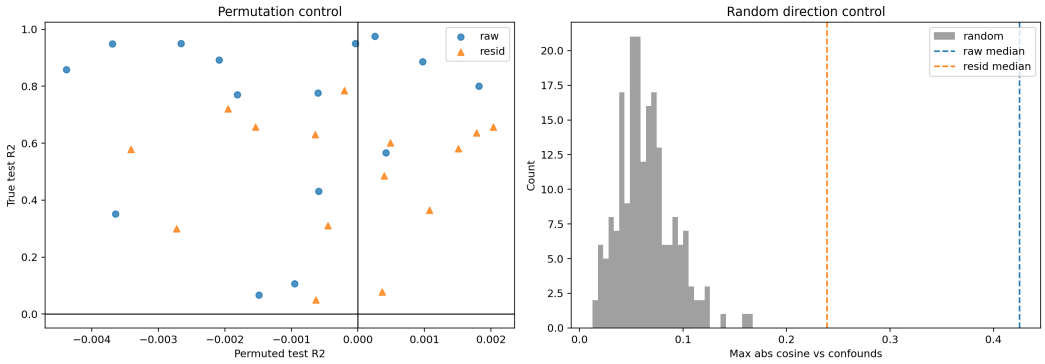

Figure 6: Control analyses. **Left:** permutation control, showing that test-set R^2 becomes approximately null when the training labels are permuted. **Right:** random-direction control, comparing the observed maximum absolute cosine similarity to confound directions against a null distribution from random latent directions.

D ADDITIONAL CONFOUND ANALYSIS

To complement the main results, we report additional analyses assessing (i) the degree to which chemical properties are associated with simple SELFIES-derived confounds and (ii) the geometric alignment between property directions and confound directions in latent space.

D.1 Correlations between properties and confounds

Figure 7 reports Pearson and Spearman correlations between the chemical properties and the confounds. The two correlation matrices are highly consistent, indicating that the observed associations are not driven solely by linear outliers, but reflect robust monotonic structure.

The strongest correlations are observed for size- and complexity-related properties. In particular, `MolWt`, `HeavyAtomCount`, and `BertzCT` are strongly positively correlated with SELFIES length and, to a lesser extent, with branch- and ring-token counts. Similarly, `RingCount` and `AromaticRingCount` show clear positive association with the ring-token count, as expected. These patterns indicate that part of the predictive signal for such properties can be recovered from relatively shallow sequence-level statistics alone.

Several additional properties, including `cLogP`, `TPSA`, `HBA`, `HBD`, and `NumRotatableBonds`, also exhibit moderate correlations with one or more confounds, suggesting partial confound mediation. By contrast, `NumSpiroAtoms` and `NumBridgeheadAtoms` remain weakly correlated overall, while `FractionCSP3` and `QED` show weaker or negative associations, especially with entropy- and length-related variables. Overall, these results support the use of residualized targets as a stricter test of whether the latent representation encodes property information beyond simple SELFIES-derived statistics.

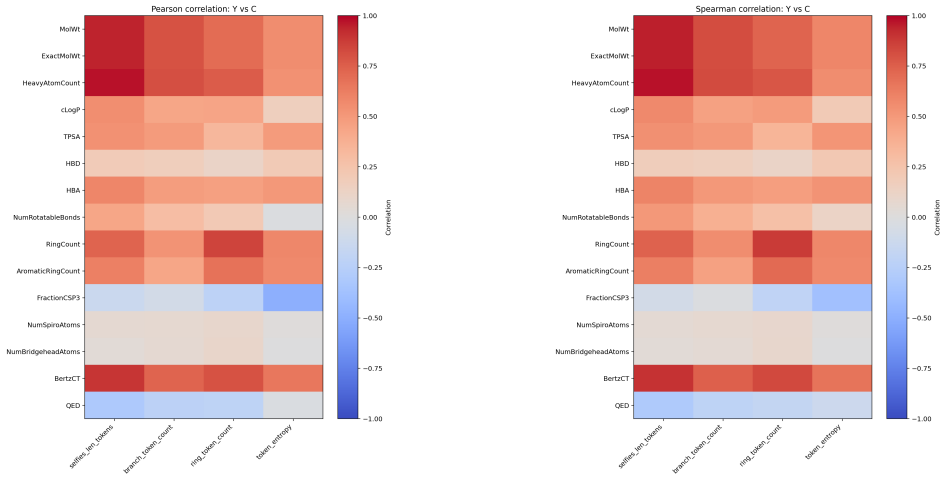

Figure 7: Correlation between molecular properties and SELFIES-derived confounds. Pearson and Spearman correlations show highly consistent structure, indicating robust monotonic associations between several properties and simple sequence-level statistics.

D.2 Cosine similarity between property and confound directions

To assess the geometry of the learned directions, we compute the cosine similarity between each property direction w_j and each confound direction v_k,

$$\cos(w_j, v_k) = \frac{w_j^\top v_k}{\|w_j\| \, \|v_k\|}. \tag{1}$$

Figure 8 shows these cosine similarities for both raw-property directions and residual-property directions. The raw heatmap reveals broad alignment between many property directions and confound directions. This is particularly pronounced for `MolWt`, `HeavyAtomCount`, and `BertzCT`, which are strongly aligned with the SELFIES length direction, and for ring-related descriptors, which align with ring-token count. These results are consistent with the correlation analyses and indicate that part of the raw latent predictability is mediated by latent axes encoding simple sequence-level statistics.

After residualization, the cosine structure is markedly attenuated and closer to zero overall. Nevertheless, several residual-property directions still retain non-negligible alignment with one or more

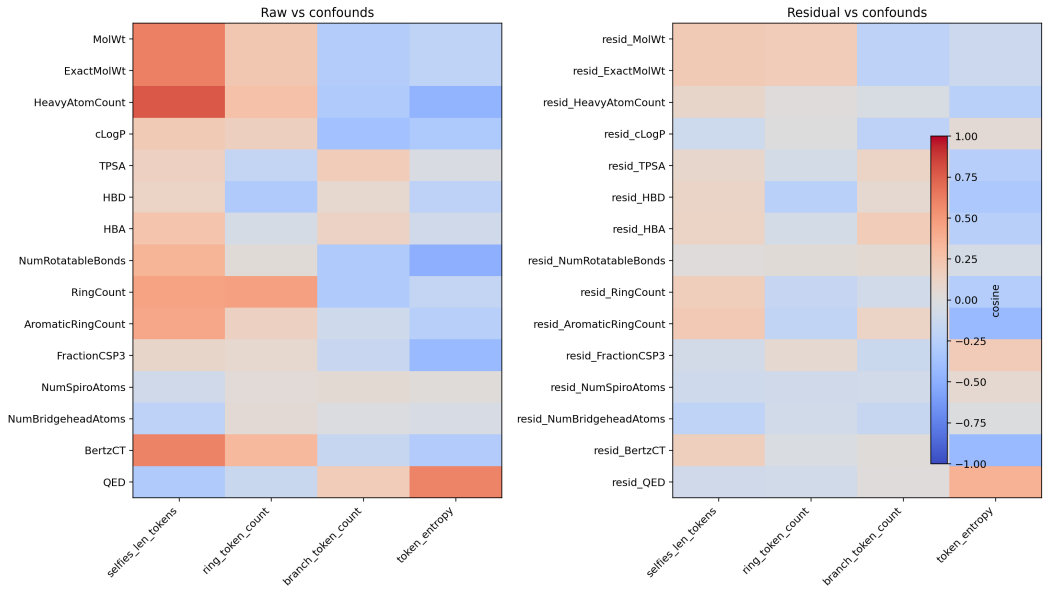

Figure 8: Cosine similarity between property directions and confound directions in latent space. **Left:** raw-property directions. **Right:** residual-property directions. Residualization reduces, but does not entirely eliminate, directional overlap with the confound subspace.

confound directions. This suggests that residualization removes the component of the target that is statistically explained by the measured confounds, but does not enforce complete geometric disentanglement in latent space. In other words, residual probes can retain genuine property-related information beyond the explicit confounds, while still occupying partially overlapping latent subspaces.

D.3 Nonlinear nuisance control

To test whether linear residualization underfits the nuisance relation, we repeat confound control with nonlinear nuisance MLPs under a leakage-controlled protocol. Training residuals use five-fold cross-fitted nuisance predictions; validation and test residuals are produced only by a nuisance model fitted on the training partition. The nuisance MLP uses ReLU hidden layers of widths 64 and 32, batch size 1,024, early stopping, and at most 75 iterations.

Table 14: Held-out latent predictability after nonlinear nuisance residualization.

Property	Nonlinear-residual latent R^2
FractionCSP3	0.761
HBA	0.662
BertzCT	0.648
cLogP	0.641
TPSA	0.641
HeavyAtomCount	0.563

Fresh nonlinear models were then trained to predict these residual targets directly from the original confound panel on held-out data. The largest remaining predictability was $R^2 = 0.0091$ for HeavyAtomCount; RingCount, ExactMolWt, and MolWt gave 0.0017, 0.0010, and 0.0009, respectively, with nearly all other values below 0.001. Thus, under this specified confound panel and nuisance-model class, the surviving latent signal is not explained by underfitting of the original linear nuisance relation. This is a robustness check against the measured SELFIES confounds, not a claim of causal disentanglement or exhaustive graph-level control.

For reference, Table 15 summarizes the descriptor regimes used in the main-text interpretation.

Table 15: Descriptor regimes from the confound-aware probe analysis.

Regime	Descriptors	Evidence
Confounded	HeavyAtomCount, MolWt, BertzCT, RingCount, AromaticRingCount	Strong correlation with SELFIES confounds; raw R^2 alone is not sufficient
Chemically meaningful residual signal	FractionCSP3, TPSA, cLogP, HBA, SA-score	High AR raw R^2 and high AR residual R^2 after confound control
Weak or likely nonlinear	HBD, QED, NumRotatableBonds, NumSpiroAtoms, NumBridgeheadAtoms	Weaker or interaction-dependent linear signal; candidates for local nonlinear analysis

E Additional Insights for Linear Probing and Monotonic Traversal

E.1 Monotonic Traversal Hypothesis

The linear probe formulation guarantees monotonic change only for the probe prediction $\hat{y}(z)$. However, the chemically relevant question is stricter: whether moving through the latent space along the learned direction produces decoded molecules whose measured property changes monotonically. This is not automatic, because the decoder is nonlinear, many-to-one, and the target property is evaluated only after decoding the latent point back into a molecular graph.

For a property direction w_y, we define the normalized traversal direction

$$u_y = \frac{w_y}{\|w_y\|_2}.$$

Given a seed molecule x_0 with latent code

$$z_0 = E(x_0),$$

we construct a one-dimensional latent path

$$z(t) = z_0 + t u_y,$$

where t controls the traversal magnitude and sign. Positive values of t correspond to moving in the direction predicted to increase the property, while negative values correspond to moving in the opposite direction.

This gives a closed-form, non-gradient optimization procedure for the linear surrogate. Indeed, within a local latent trust region

$$\mathcal{B}_\rho(z_0) = \{z : \|z - z_0\|_2 \le \rho\},$$

maximizing the probe prediction

$$\hat{y}(z) = w_y^\top z + b$$

has the analytic solution

$$z^\star = z_0 + \rho \frac{w_y}{\|w_y\|_2}.$$

Candidate molecules are obtained by evaluating a one-dimensional path along u_y, rather than solving a high-dimensional nonlinear optimization problem.

Each point $z(t)$ is decoded into a SELFIES sequence and converted back into a molecular graph. The target property is then recomputed from the decoded molecule using the same RDKit descriptor used to construct the property panel. This gives an observed property trajectory

$$y(t) = g(D(z(t))),$$

where D denotes decoding and g denotes the graph-based property function. For stochastic decoding, multiple samples may be generated at each traversal step and summarized by the mean property among valid decoded molecules,

$$\bar{y}(t) = \frac{1}{|\mathcal{V}_t|} \sum_{x \in \mathcal{V}_t} g(x),$$

where $\mathcal{V}_t$ is the set of valid decoded molecules at step t.

The monotonic traversal hypothesis is that, for chemically meaningful global directions, the decoded property trajectory should vary consistently with the traversal coordinate:

$$t_1 < t_2 \quad \Longrightarrow \quad \bar{y}(t_1) \le \bar{y}(t_2)$$

for increasing directions, up to stochastic decoding noise and local chemical constraints. In practice, this hypothesis is evaluated using rank correlation between t and $\bar{y}(t)$, the slope of a linear fit along the traversal path, and the number of monotonicity violations between successive traversal points.

This test is stricter than held-out probe performance. A high probe R^2 only shows that the property is linearly decodable from the latent representation. A successful monotonic traversal shows that the learned direction is also actionable: moving through latent space changes the decoded molecular distribution in the intended chemical direction. Conversely, if probe performance is high but traversal is non-monotonic or chemically unstable, then the direction may encode property information without providing a reliable steering axis.

The same formulation also supports multi-criterion latent exploration. If several properties admit stable directions, then optimization can be performed in the low-dimensional subspace spanned by their corresponding vectors. For example, one may move toward increasing drug-likeness while penalizing synthetic inaccessibility, or search for regions where a target property, synthesizability, and functional-family retention are jointly satisfied. In this setting, property steering becomes a geometric operation in latent space: directions define axes of molecular variation, and their intersections define candidate regions where multiple chemical criteria are simultaneously favored.

We therefore use monotonic traversal as the operational validation of a global property direction. A direction is considered chemically meaningful only if it combines three properties: predictive strength in latent space, robustness to confound controls, and monotonic behavior after decoding. Additional traversal diagnostics such as validity, uniqueness, similarity to the seed molecule, and family retention are used to determine whether the property change occurs through coherent local chemical edits rather than arbitrary jumps in molecular structure.

E.2 Residual-direction traversal

We additionally traverse directions obtained by fitting the same linear probe to residualized property targets. For each of 50 held-out seeds, raw and residual directions are ℓ_2-normalized and evaluated over the same signed traversal range (up to $|\alpha| = 150$), using all valid decoded points. Table 16 reports median per-seed Spearman monotonicity.

Table 16: Raw and residual-direction traversal. Columns report the raw property along the raw direction, the raw property along the residual direction, and the residual target along the residual direction.

Property	Raw → raw	Residual → raw	Residual → residual
FractionCSP3	0.938	0.912	0.893
HBA	0.955	0.929	0.881
TPSA	0.930	0.915	0.865
cLogP	0.935	0.959	0.926
BertzCT	0.944	0.689	0.717
HeavyAtomCount	0.973	0.661	0.746

Residualization also reduces alignment with the measured confound subspace for several strongly coupled properties: HBA decreases from approximately 0.604 to 0.207, cLogP from 0.604 to 0.120, BertzCT from 0.911 to 0.293, and HeavyAtomCount from 0.975 to 0.127. The residual directions therefore preserve systematic decoded property ordering while reducing measured confound burden, particularly for FractionCSP3, HBA, TPSA, and cLogP. The weaker residual monotonicity of BertzCT and HeavyAtomCount is consistent with their stronger structural coupling. We interpret this as confound-robust directional signal, not as evidence of pure or fully disentangled latent factors.

E.3 Additional monotonic traversals.

In addition to the properties shown in the main analysis of Section 4.5, we observed two further properties with clear monotonic behavior under latent traversal: BertzCT and HeavyAtomCount. Figure 9 highlights that both traversals show a broadly increasing median trend across the latent direction, with the interquartile range indicating that the effect is preserved across the 50 decoded molecules rather than being driven by a small number of outliers.

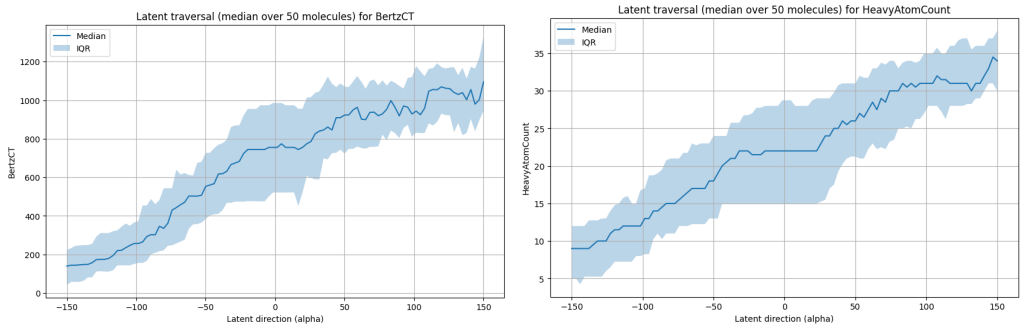

Figure 9: Additional monotonic latent traversals from Section 4.5. BertzCT and HeavyAtomCount both show broadly increasing median trends across the latent direction, with shaded regions indicating the interquartile range over 50 molecules.

E.4 Inter-Property Correlation and Directional Similarity

To better interpret the traversal results, we compare two complementary objects: the empirical correlation matrix of molecular descriptors and the cosine-similarity matrix between the corresponding linear-probe directions. The first matrix measures how properties co-vary in the dataset, while the second measures whether the latent directions recovered by the probes are geometrically aligned. Both matrices are shown in Figure 10.

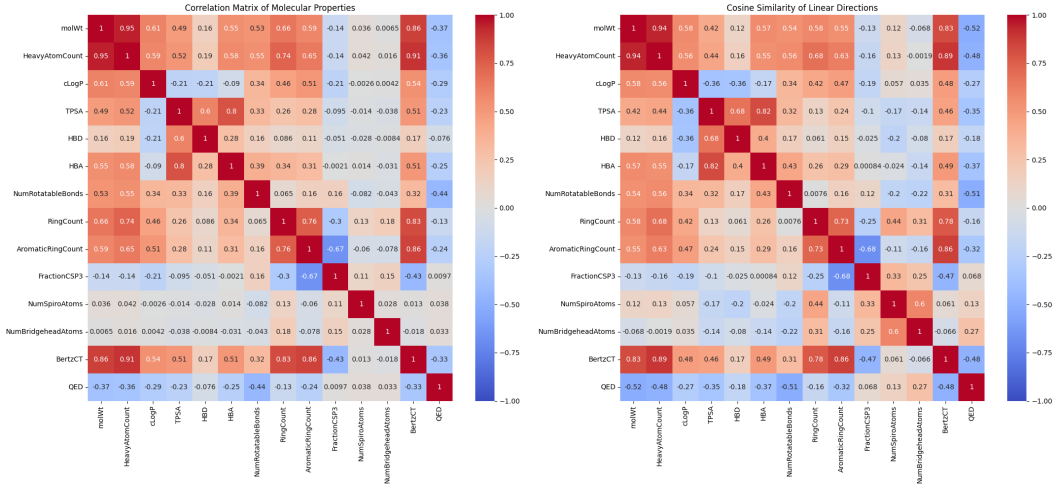

Figure 10: Inter-property structure in descriptor space and latent-direction space. Left: empirical Pearson correlation matrix between molecular properties. Right: cosine-similarity matrix between the corresponding linear-probe directions. The broad agreement between the two matrices indicates that the latent geometry preserves meaningful chemical relationships between properties, while discrepancies reveal higher-order organization not fully captured by marginal property correlations.

Figure 10 shows that several chemically expected descriptor blocks are preserved in the latent geometry. Size-related descriptors are strongly coupled: `molWt` and `HeavyAtomCount` have empirical correlation 0.95 and directional cosine similarity 0.94, while `HeavyAtomCount` and `BertzCT` have correlation 0.91 and directional similarity 0.89. Polar descriptors also form a coherent block, with `TPSA` and `HBA` showing correlation 0.80 and directional similarity 0.82. Similarly, ring and aromatic descriptors are strongly aligned: `RingCount`, `AromaticRingCount`, and `BertzCT` are mutually correlated, and their probe directions are also highly similar. In particular, `AromaticRingCount` and `BertzCT` show both empirical correlation and directional similarity of approximately 0.86.

This analysis clarifies the interpretation of the confounded descriptors. HeavyAtomCount and especially BertzCT are indeed strongly associated with size-related quantities, but their signal is not reducible to a trivial sequence-length effect. BertzCT aligns not only with molWt and HeavyAtomCount, but also with ring and aromatic descriptors, with directional similarities of 0.78 with RingCount and 0.86 with AromaticRingCount. This is consistent with BertzCT acting as a topological complexity descriptor: it combines information about molecular size, branching, ring structure, and aromatic organization. Therefore, the confounding observed for BertzCT should not be interpreted as a simple token-count artifact, but as a structured relationship between sequence-level statistics and chemically meaningful molecular topology.

The behavior of FractionCSP3 further supports this interpretation. It is negatively associated with aromaticity and topological complexity, with correlations of -0.67 with AromaticRingCount and -0.43 with BertzCT. The corresponding latent directions show nearly identical antagonism, with cosine similarities of -0.68 and -0.47, respectively. Thus, the learned geometry captures the expected opposition between saturated three-dimensional character and aromatic or topologically complex structure.

Some topological descriptors also show stronger directional alignment than raw descriptor correlation. For example, NumSpiroAtoms and NumBridgeheadAtoms have weak empirical correlation (0.028), but their latent directions have cosine similarity 0.60. Likewise, RingCount is only weakly correlated with NumSpiroAtoms (0.13), while their latent directions are more clearly aligned (0.44). This suggests that the model groups rare ring-topology features in latent space even when they are not strongly correlated at the marginal descriptor level.

Overall, Figure 10 shows that the learned directions are chemically structured rather than independent one-property artifacts. The latent geometry broadly mirrors known descriptor relationships while also revealing higher-order organization among topological features. This supports the interpretation that the successful traversals of HeavyAtomCount and BertzCT, despite their confound sensitivity, arise from a richer latent encoding of molecular topology and complexity rather than from a purely superficial encoding of SELFIES length or token frequency.

E.5 Ridge vs Linear Regression Ablation Study

The main linear-probe study uses ordinary linear regression rather than ridge regression because the fitted probe vector is not used only as a predictive readout. In our setting, the learned weight vector w is also reused as an actionable latent direction for traversal and decoded property steering. This makes the geometry of w part of the method itself.

Ridge regression was initially considered because it often gives strong and stable probe performance in high-dimensional latent spaces. Given latent vectors z_i and target property values y_i, ridge regression solves

$$\min_{w,b} \sum_i (y_i - w^\top z_i - b)^2 + \lambda \|w\|^2,$$

where w is the learned linear coefficient vector, b is the bias term, i indexes the training examples, and $\lambda \geq 0$ is the regularization strength. The first term measures the squared prediction error, while the second term penalizes large coefficient norms.

This penalty can improve predictive robustness by discouraging overly large weights. However, the same penalty is not geometrically neutral when w is interpreted as a steering direction in latent space: it may shrink or rotate the fitted vector relative to the unregularized least-squares solution. For this reason, ridge regression is better interpreted here as a regularized predictive probe, whereas ordinary Linear Regression is the cleaner choice for the main steering analysis.

We therefore distinguish the two probes as answering related but different questions:

$$\text{Ridge: regularized predictive readability},$$

$$\text{Linear Regression: unregularized direction for steering}.$$

This does not make ridge inappropriate. On the contrary, ridge remains useful when predictive accuracy, stability, or property readout is the primary objective. We include it as a complementary sensitivity analysis to verify that the latent-property conclusions are not specific to the unregularized probe.

Table 17: Ridge versus ordinary Linear Regression for the autoregressive MultiSlotting model. Both raw and residual R^2 are reported. Δ denotes ridge minus Linear Regression.

Property	Linear raw	Ridge raw	Δ_{raw}	Linear resid.	Ridge resid.	Δ_{resid}
HeavyAtomCount	0.9750	0.9884	+0.0134	0.6554	0.7692	+0.1138
MolWt	0.9480	0.9655	+0.0175	0.5918	0.7000	+0.1082
BertzCT	0.9512	0.9677	+0.0165	0.7009	0.7537	+0.0528
RingCount	0.9378	0.9567	+0.0189	0.7944	0.8506	+0.0562
AromaticRingCount	0.8937	0.9090	+0.0153	0.7559	0.7767	+0.0208
cLogP	0.7967	0.8442	+0.0475	0.6861	0.7552	+0.0691
TPSA	0.8059	0.8670	+0.0611	0.6863	0.7728	+0.0865
HBA	0.8008	0.8389	+0.0381	0.6759	0.7138	+0.0379
FractionCSP3	0.8951	0.9153	+0.0202	0.8313	0.8542	+0.0229
HBD	0.4997	0.5535	+0.0538	0.4624	0.5211	+0.0587
NumRotatableBonds	0.6914	0.7640	+0.0726	0.4752	0.5960	+0.1208
NumSpiroAtoms	0.0878	0.1073	+0.0195	0.0728	0.0914	+0.0186
NumBridgeheadAtoms	0.1522	0.1878	+0.0356	0.1207	0.1556	+0.0349
QED	0.4624	0.5121	+0.0497	0.3907	0.4424	+0.0517
Mean	0.7070	0.7412	+0.0343	0.5643	0.6252	+0.0609

Table 17 shows that, as expected, ridge generally improves predictive R^2, with an average gain of 0.0343 in the raw setting and 0.0609 after residualization. The gain is most visible for residualized size- or topology-related descriptors such as HeavyAtomCount, MolWt, and NumRotatableBonds, where regularization stabilizes the readout. Importantly, however, the global interpretation is unchanged. FractionCSP3, TPSA, cLogP, and HBA remain strongly predictable under both probes, while NumSpiroAtoms and NumBridgeheadAtoms remain weak under linear readout. Thus, ridge improves predictive readability but does not alter the qualitative distinction between globally steerable descriptors and descriptors that likely require nonlinear or local structure.

Table 18: Ridge-probe comparison across the three latent models. Raw and residual R^2 are reported for the same descriptor panel used in the previous main-text analysis. Best values within each target setting are bolded.

Property	Linear Attn. raw	Simple Attn. raw	AR raw	Linear Attn. resid.	Simple Attn. resid.	AR resid.
HeavyAtomCount	0.9759	0.9622	**0.9884**	0.4857	0.3787	**0.7692**
MolWt	0.9502	0.9190	**0.9655**	0.5808	0.3170	**0.7000**
BertzCT	0.9494	0.9303	**0.9677**	0.6564	0.5646	**0.7537**
RingCount	0.8918	0.8495	**0.9567**	0.6013	0.5154	**0.8506**
AromaticRingCount	0.8855	0.8503	**0.9090**	0.7210	0.6577	**0.7767**
cLogP	0.7699	0.6824	**0.8442**	0.6572	0.5791	**0.7552**
TPSA	0.7765	0.7008	**0.8670**	0.6307	0.5251	**0.7728**
HBA	0.7998	0.7602	**0.8389**	0.6360	0.5427	**0.7138**
FractionCSP3	0.8587	0.7862	**0.9153**	0.7843	0.7161	**0.8542**
HBD	0.3518	0.2627	**0.5535**	0.3100	0.2226	**0.5211**
NumSpiroAtoms	0.0666	0.0513	**0.1073**	0.0504	0.0358	**0.0914**
NumBridgeheadAtoms	0.1070	0.0816	**0.1878**	0.0781	0.0490	**0.1556**
QED	0.4308	0.3455	**0.5121**	0.3652	0.2789	**0.4424**
Mean	0.6780	0.6294	**0.7395**	0.5044	0.4141	**0.6274**

Moreover, the ridge comparison in Table 18 confirms that the autoregressive MultiSlotting model remains the strongest representation even under a regularized predictive probe. It obtains the best raw and residual R^2 across the descriptor panel, including both confound-sensitive descriptors such as HeavyAtomCount, MolWt, and BertzCT, and chemically meaningful residual descriptors such as FractionCSP3, TPSA, cLogP, and HBA. This supports the same qualitative conclusion as the main Linear Regression analysis: the autoregressive latent space is more readable and better organized than the ablation models.

Overall, the ridge ablation reinforces the robustness of the latent interpretability results while clarifying the methodological role of the main probe. Ridge is useful as a prediction-oriented sensitivity analysis, but the main steerability claims are based on ordinary Linear Regression because the learned direction is used directly for latent traversal and decoded molecular steering.

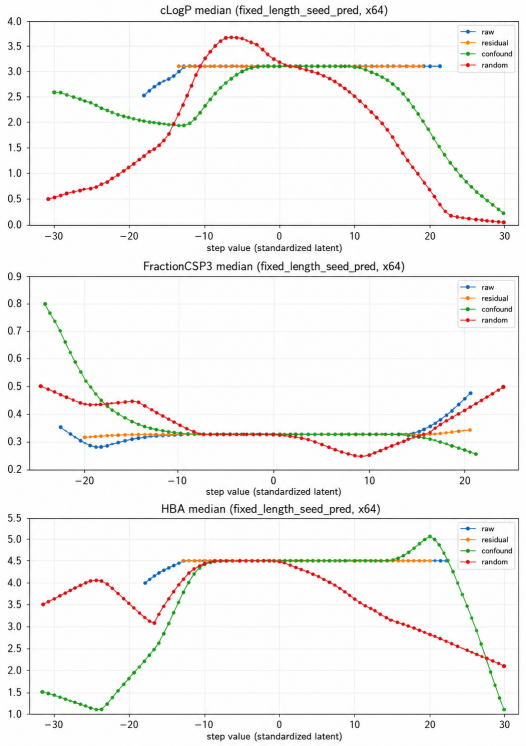

Figure 11: Raw, confound, residual, and random-axis latent traversals for the linear-attention model. The decoded properties are mostly flat near the seed and exhibit abrupt, saturated, or non-monotonic changes rather than stable directional control.

E.6 Simple and linear attention models and their latent traversals

For completeness, we also evaluated latent traversals for the linear-attention and simple-attention baselines. These experiments are useful because they test a stricter condition than probe accuracy: not only whether a property is decodable from the latent representation, but whether the corresponding probe direction behaves like a usable control axis after decoding back to molecules.

Figure 11 shows representative traversals for the linear-attention model on cLogP, F.CSP3, and HBA. The dominant pattern is not a smooth property gradient, but a combination of plateaus, cliffs, and saturation. Around the seed, decoded molecules often remain effectively unchanged over a wide range of step sizes, suggesting that movement along the latent direction is absorbed by the decoder without producing chemically distinct outputs. When the property does change, the transition is frequently abrupt or non-monotonic, and the targeted raw direction is not cleanly separated from residual, confound, or random directions. In other words, the property signal may be present in the latent state, but it is not organized as a stable, axis-aligned direction that supports controlled molecular editing.

Figure 12 shows the corresponding raw-axis traversals for the simple-attention model. This baseline is more responsive: cLogP, F.CSP3, and HBA all show clear property movement along the raw probe direction. However, the response is still not consistently traversal-ready. F.CSP3 rapidly increases and then saturates, indicating that the direction captures a coarse transition rather than a continuously tunable property. HBA shows a large increase over part of the traversal but later bends back downward, while cLogP exhibits pronounced scale dependence and non-monotonic behavior. Thus, the simple-attention latent space contains stronger chemically meaningful signal than the linear-attention model, but that signal is only locally useful and does not define a globally reliable control coordinate.

Together, these baselines highlight the distinction between *latent decodability* and *latent controllability*. A probe can recover property information from a representation even when moving along

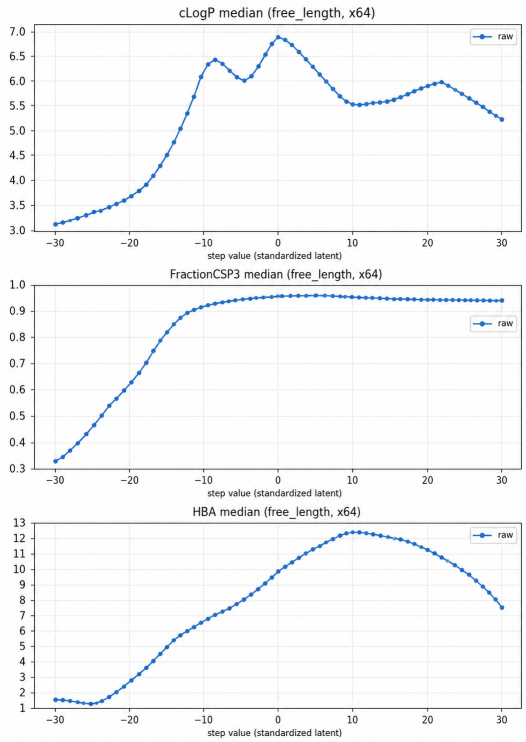

Figure 12: Raw-axis latent traversals for the simple-attention model. The simple-attention baseline produces clearer property responses than the linear-attention model, but the trajectories show saturation, non-monotonicity, and dependence on traversal scale.

its direction does not induce a smooth or monotonic change in decoded molecules. The attention baselines therefore encode chemically relevant information, but their latent geometry is less aligned with direct molecular control than the autoregressive model. In the autoregressive setting, probe directions behave more like usable editing axes; in the attention baselines, they behave more like correlational readout directions whose effect on decoded chemistry is discontinuous, saturated, or strongly scale-dependent.

F Non-Linear Probing

For completeness, Table 19 reports the full expanded-property MLP comparison used in the main-text nonlinear-probe analysis.

Table 19: MLP probe performance on the expanded property set for the autoregressive and simple-attention models, before and after confound residualization.

Model / target	HAC	MolWt	BertzCT	Ring	A.Ring	cLogP	TPSA	HBA	F.CSP3	HBD	Spiro	Bridge	QED	NumRotBond	SA	Mean
AR MLP raw	**0.996**	**0.990**	**0.986**	**0.985**	**0.955**	**0.934**	**0.946**	**0.926**	**0.982**	**0.792**	**0.628**	**0.785**	**0.803**	**0.909**	**0.814**	**0.895**
Simple-attention MLP raw	0.990	0.979	0.975	0.953	0.933	0.890	0.905	0.905	0.959	0.644	0.436	0.680	0.685	0.821	0.711	0.831
AR MLP residual	**0.933**	**0.938**	**0.917**	**0.958**	**0.902**	**0.901**	**0.916**	**0.884**	**0.966**	**0.778**	**0.608**	**0.771**	**0.771**	**0.849**	**0.795**	**0.859**
Simple-attention MLP residual	0.830	0.834	0.851	0.855	0.857	0.837	0.842	0.834	0.930	0.626	0.423	0.656	0.640	0.701	0.671	0.759

F.1 Gradient-Based Local Directions

Unlike the linear probe, the MLP does not define a single global direction. Its local direction at a latent point is given by the gradient of the learned predictor,

$$g_y(\tilde{z}) = \nabla_{\tilde{z}} f_{\phi,y}(\tilde{z}),$$

where $\tilde{z}$ denotes the standardized latent vector, $f_{\phi,y}$ is the MLP predictor for property y with learned parameters ϕ, and $g_y(\tilde{z})$ is the gradient of that predictor with respect to $\tilde{z}$.

The corresponding normalized local steering direction is

$$u_y(\tilde{z}) = \frac{g_y(\tilde{z})}{\|g_y(\tilde{z})\|_2 + \varepsilon},$$

where $u_y(\tilde{z})$ is the unit-normalized local direction, $\|\cdot\|_2$ denotes the Euclidean norm, and $\varepsilon > 0$ is a small constant used to avoid division by zero.

A nonlinear traversal is therefore defined recursively as

$$\tilde{z}_{k+1} = \tilde{z}_k + \eta u_y(\tilde{z}_k),$$

where k indexes the traversal step and $\eta > 0$ is the step size. Traversal in the decreasing direction is obtained by replacing $+\eta$ with $-\eta$. Since the MLP is trained in standardized latent coordinates, the traversed point is mapped back to the decoder space by

$$z_k = \mu_Z + s_Z \odot \tilde{z}_k,$$

where z_k is the corresponding latent vector in the decoder coordinate system, μ_Z and s_Z are the empirical mean and standard deviation of the latent vectors used for standardization, and $\odot$ denotes elementwise multiplication.

This makes nonlinear steering a path through a learned vector field rather than a straight-line displacement. The linear probe gives

$$\nabla_z(w^\top z + b) = w,$$

where w and b are the linear probe coefficient vector and bias term. Thus, every molecule shares the same direction. The MLP gives

$$\nabla_{\tilde{z}} f_\phi(\tilde{z}),$$

where f_ϕ denotes the learned nonlinear predictor, so the direction depends on the current molecule. Local steering is therefore most appropriate for properties that are predictable from z but not organized around a stable global axis.

As with linear traversal, decoded molecules along the path are evaluated using the external RDKit property function,

$$y_k = g(D(z_k)),$$

where D is the trained decoder, $D(z_k)$ is the molecule decoded from the latent vector z_k, g is the external RDKit property evaluator, and y_k is the resulting property value at traversal step k. The MLP gradient is therefore treated as a steering hypothesis whose chemical validity must be verified after decoding.

F.1.1 Interpreting Delta-R^2

To distinguish globally linear organization from merely decodable information, we define

$$\Delta R^2 = R^2_{\text{MLP}} - R^2_{\text{linear}}.$$

Small values indicate that a linear probe already captures most of the available property signal, whereas large values indicate additional nonlinear structure accessible only to the MLP.

Table 20 reports the Delta-R^2 analysis on the expanded property set. This diagnostic supports the traversal results. The descriptors that showed monotonic behavior under linear latent traversal; HeavyAtomCount, BertzCT, FractionCSP3, TPSA, cLogP, and HBA, also have comparatively low ΔR^2. In the autoregressive model, their mean ΔR^2 is 0.061 in the raw setting and 0.152 after residualization. By contrast, the nonlinear descriptor group {HBD, NumRotatableBonds, NumSpiroAtoms, NumBridgeheadAtoms, QED} has much larger mean gaps, 0.360 raw and 0.396 residual. The same trend appears in the simple-attention model, but with larger gaps for the monotonic descriptors, increasing to 0.134 raw and 0.305 residual, and a much larger gap for the nonlinear descriptor (0.410 raw and 0.453 residual).

Thus, both models encode chemically meaningful descriptor information, but the autoregressive latent space organizes the traversal-relevant descriptors in a more globally linear way. This explains why these properties are more stable under linear latent traversal. AromaticRingCount is excluded from the clean monotonic group because, although its ΔR^2 is small, the confound analysis showed

substantial correlation with ring-token features. SA is also not treated as a clean traversal target because its traversal behavior is unstable under abrupt structural changes.

Table 20: Delta-R^2 analysis on the expanded property set. Low ΔR^2 indicates that the linear probe already captures most of the available signal; high ΔR^2 indicates substantial nonlinear structure.

Property	Regime / interpretation	AR ΔR^2_{raw}	AR $\Delta R^2_{\text{resid}}$	Simple ΔR^2_{raw}	Simple $\Delta R^2_{\text{resid}}$
F.CSP3	monotonic linear regime	0.067	0.111	0.175	0.220
TPSA	monotonic linear regime	0.078	0.143	0.206	0.319
cLogP	monotonic linear regime	0.089	0.145	0.207	0.260
HBA	monotonic linear regime	0.104	0.180	0.146	0.294
HAC	monotonic linear regime (size confounded)	0.007	0.163	0.028	0.449
BertzCT	monotonic linear regime (size confounded)	0.021	0.169	0.044	0.289
MolWt	confounded regime (size confounded)	0.024	0.235	0.062	0.523
Ring	confounded regime (ring confounded)	0.029	0.107	0.104	0.340
A.Ring	confounded regime (ring confounded)	0.047	0.126	0.083	0.205
HBD	nonlinear structure	0.237	0.256	0.385	0.406
Spiro	nonlinear structure	0.517	0.513	0.385	0.388
Bridge	nonlinear structure	0.592	0.610	0.603	0.612
QED	nonlinear structure	0.307	0.345	0.339	0.359
NumRotBond	nonlinear structure	0.145	0.253	0.337	0.502
SA	unstable traversal target	0.160	0.180	0.254	0.286
Mean over monotonic descriptors	HAC, BertzCT, F.CSP3, TPSA, cLogP, HBA	0.061	0.152	0.134	0.305
Mean over nonlinear descriptors	HBD, Spiro, Bridge, QED, NumRotBond	0.360	0.396	0.410	0.453