MULTIUS : UNSUPERVISED PERSONA MIXTURE AND NORM-BOUNDED LATENT STEERING FOR USER SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Obtaining a faithful User Simulator is a precondition for effective evaluation and training of multi-turn conversational systems. Prompt-based user simulators, however, reproduce human utterances and behavior poorly, while fine-tuning approaches require substantial computational resources. Recently, steering vectors have emerged as a promising and cost-effective mechanism for inducing target behaviors in Large Language Models, and have been successfully applied to persona and behavior control. However, computing steering vectors typically relies on curated contrastive samples and therefore requires prior knowledge of the relevant behavioral dimensions and their intensities.

In this work, we propose to infer the distribution of user behaviors directly from real conversations, using unsupervised Gaussian-Mixture models in a latent behavioral space. Samples from these distributions are mapped to steering vectors through lightweight adapter layers applied to a frozen Language Model. The method allows to learn a single User Simulator, trained once, and subsequently generate K distinct user behaviors by sampling from the learned distributions, without requiring explicit persona labels or retraining.

To preserve the base model's activation geometry, we constrain the steering operation so that, at every token, the steered activation norm remains within $(1\pm\tau)$ of the original norm. We study two adapter families: a per-channel affine map in the spirit of AdaLN-Zero, and a parameter-efficient variant operating entirely in a low-dimensional subspace. On WildChat, a single simulator adding 7.9–76.3M trainable parameters (0.2–1.8% of the frozen Qwen3.5-4B) writes far more human-like messages than a fully fine-tuned 8B user simulator and than prompted open-weight and frontier models (measured through relative MAUVE scores), is closest to human register, and stops conversations when humans do. On PRISM, a population it was never trained on, it is about as close to real users as real WildChat users are. We name our approach MultiUS.

1 INTRODUCTION

User simulation has become an essential tool in the training and evaluation of modern conversational systems. Conversational agents are multi-turn decision processes, where a policy acts on environments over many turns (Zhang et al., 2025) : evaluating those systems on static benchmarks therefore does not expose them to this setting, and models that perform well in static benchmarks often degrade in multi-turn conversation (Laban et al., 2025). Interaction-based benchmarks therefore leverage User Simulation to evaluate LLM performance on multi-turn scenarios (Yao et al., 2024; Barres et al., 2026; Shi et al., 2026), but assistant performance is highly sensitive to the simulator (Mehri et al., 2026). As a result, a useful user simulator must behave like real users, and must reflect their diversity. Due to its simplicity, user simulators are commonly implemented by prompting assistant models, but such simulators fail to capture human behavior (Mehri et al., 2026), biasing evaluation and overstating the model's capabilities (Seshadri et al., 2026). Fine-tuning on real user turns has been shown to close this gap (Naous et al., 2025; Sun et al., 2024), but it learns user populations as a single distribution, offering no control over the user profile, without knowing them in advance.

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.

Real world user populations include subpopulations of users with distinct behaviors, and different sub-groups might react differently to the exact same system responses. An assistant evaluated against a single "average" user measures performance on an user who may not exist, and during training, a policy optimized against a single simulator can overfit to its idiosyncrasies, whereas a diverse set of interaction partners is a standard remedy against such exploitation (Liu et al., 2023). As a counter measure, recent work conditions simulators on user profiles (Wang et al., 2025; Paglieri et al., 2026; Chopra et al., 2026). These profiles, however, are specified explicitly, written or generated in natural language or extracted by LLM annotators under a predefined schema, and must therefore be known, or at least describable, in advance.

Activation steering has since emerged as a cheap and effective alternative by steering a frozen model's behavior at inference time by intervening on its hidden states (Rimsky et al., 2024; Wu et al., 2024; Li et al., 2023; Rodriguez et al., 2025; He et al., 2026). However, a fundamental obstacle for applying these methods for user simulation is that they require the target behavior to be specified, through contrastive pairs, labeled examples or target distributions (Rodriguez et al., 2025). However, for users, these behaviors vary across populations and are typically unknown.

In this work, we introduce MultiUS, a variational autoencoder with a Gaussian mixture prior, following VaDe (Jiang et al., 2017) and GM-VAE (Dilokthanakul et al., 2017), whose decoder is a frozen language model steered by lightweight adapters, and whose latent variable determines the steering vector to be added to the model's activations. The mixture components foregoes the curation of behavior demonstrations and discovers user behaviors discovered without supervision, while the continuous latent provides smooth control within and across them. We apply this setting to user simulation and present a gated activation steering mechanism with the objective of cheaply learning a single User Simulator, trained once, and capable of reproducing K distinct learned user behaviors, while adding relatively few parameters to a frozen small language model. To preserve the base model's activation geometry, we constrain the steering operation so that, at every token, the steered activation norm remains within $(1\pm\tau)$ of the original norm. In this work, our contributions are :

- A norm-bounded activation steering mechanism for frozen language models, driven by a latent space of user profiles learned without supervision. We study two adapter families: a per-channel affine map in the spirit of AdaLN-Zero, and a parameter-efficient variant operating entirely in a low-dimensional subspace.
- A user simulator built on this mechanism that reproduces K distinct behavior profiles from a single model without retraining, adding only 7.9M to 76.3M parameters to a frozen Qwen3.5-4B backbone.
- On PRISM, which neither MultiUS nor its training data has seen, our simulator's first messages score a MAUVE of 0.105 [0.091, 0.124], within the confidence interval of real WildChat users scored against the same corpus (0.112 [0.093, 0.135]), the best a simulator trained on WildChat can be expected to achieve. On held-out WildChat, MultiUS reaches 0.676 [0.646, 0.705] against a real-user ceiling of 0.964, while UserLM-8b scores 0.066 [0.055, 0.074]. The low-rank variant, with 9.65$\times$ fewer trainable parameters, scores 0.615 [0.574, 0.652].

2 Related Work

Variational Text Generation Variational autoencoders for text learn a continuous latent space from which sentences are decoded (Bowman et al., 2016). Works such as OPTIMUS (Li et al., 2020) have scaled those approaches by pairing pretrained encoders and decoders using a simple Gaussian prior. A persistent difficulty is posterior collapse, where a strong autoregressive decoder successfully predicts sentences while ignoring the latent (Bowman et al., 2016), and is typically mitigated with cyclical KL annealing (Fu et al., 2019) and freebits thresholds (Kingma et al., 2016). DEM-VAE (Shi et al., 2020) successfully applied mixture priors introduced by VaDe (Jiang et al., 2017) and GM-VAE (Dilokthanakul et al., 2017), while addressing mode collapse. These models inject the latent through the input embeddings, attention memory or prefixes, and usually fine-tune the decoder. We build on the VaDE objective, with an amortized cluster posterior and an entropy regularizer against component collapse, and depart in how the latent reaches the decoder: the language model remains entirely frozen, and the latent acts only through bounded modulations of its intermediate activations.

User simulation Fine-tuning on real user turns substantially narrows the gap to human behavior (Naous et al., 2025; Sun et al., 2024), but is costly and models the population as a single distribution, with no control over which user is simulated. Multi-profile simulators instead condition on user profiles. USP (Wang et al., 2025) fine-tunes a simulator on implicit profiles extracted by an LLM under predefined rubrics. mTAD (Ferreira et al., 2024) trains LoRA (Hu et al., 2021) adapters on trait-isolated datasets, curated with handcrafted trait identification, and combines them at inference, so its cost grows linearly with the number of traits, whereas a single set of our adapters serves all K profiles. Persona Generators (Paglieri et al., 2026) and Persona Policies (Chopra et al., 2026) optimize persona generators by LLM-driven evolutionary search and prompt an LLM with their outputs, inheriting the limitations of prompt-based simulators (Mehri et al., 2026; Seshadri et al., 2026). In all these approaches, profiles are specified in advance: written, generated, or extracted along predefined shemes or axes.

Activation Steering Most existing methods compute steering vectors to be added to the model's hidden states during decoding. ITI (Li et al., 2023) and CAA (Rimsky et al., 2024) compute fixed steering vectors from contrastive pairs, and apply the same update across inputs. Similar works like Trait-basis (He et al., 2026) and Persona Vectors (Chen et al., 2025) apply fixed steering vectors to behavior control. On the other hand ReFT (Wu et al., 2024) steers model behavior by applying learned, low-rank transformations directly to the activation values at specific architectural layers and token positions. LinACT (Rodriguez et al., 2024) computes its steering vector from affine transformations of the token hidden representations, and LinEAS (Rodriguez et al., 2025) builds on top of this mechanism, optimizing the entirety of the steering adapters end-to-end with a global loss function.

Activation steering has been shown to risk degrading the steered model's performance (Taimeskhanov et al., 2026). Promising methods address this concern by adapting the intervention to the input, deciding when or how strongly to steer (Ferrando et al., 2026; Hedström et al., 2025; Lee et al., 2024), or constraining it to leave unrelated inputs unaffected (Sheng et al., 2025). A recent line of work performs norm-preserving activation steering, ensuring the perturbed activations remain on the surface of the hypersphere defined by the original activation norm (Ngo et al., 2026; You et al., 2026). As a design choice, we choose a relaxation of this constraint, as will be discussed in Section 3.

Existing methods requires samples of the desired behavior, which, for user simulation, is often unknown and varies according to the user population. Our steering is instead conditioned on a latent variable inferred from data, so no target behavior is specified at any stage.

3 METHOD

3.1 PROBLEM FORMULATION

We model user utterances u_i from contexts c_i with a latent profile variable $z \in \mathbb{R}^{d_z}$ whose prior is a mixture of K Gaussians indexed by y:

$$p(y = k) = \pi_k, \quad p(z|y = k) = \mathcal{N}(\mu_k, \mathrm{diag}(\sigma_k^2)), \quad p_\theta(u|c, z)$$

This is a VAE with a Gaussian-mixture prior. The profile structure is discovered without supervision, while the continuous latent allows smooth modulation of the frozen backbone.

Backbone Let p_{base} be a frozen LLM with L decoder layers $f_1, \ldots, f_L$ acting on a token sequence x as $h_0 = \mathrm{Emb}(x) \in \mathbb{R}^{T\times d}$, $h_l = f_l(h_{l-1})$, $\mathrm{logits} = \mathrm{LM}(h_L)$, with $\mathrm{LM}(\cdot)$ the Language Model head and $\mathrm{Emb}(\cdot)$ the Embedding layer.

Decoder We steer p_{base} by installing adapters $T_l^*, * \in \{\mathrm{att}, \mathrm{ffn}\}$, at a set of install sites $\mathcal{S} \subseteq \{1, \ldots, L\}$ following similar notation to Ferrando et al. (2026). A typical LLM decoder layer contains two normalization modules, one preceding the attention block and one preceding the FFN. At every install site $l \in \mathcal{S}$, we wrap both with T_l^{att} and T_l^{ffn} respectively, each applying the steering mechanism of Section 3.2 to the module's input. The two adapters are structurally identical, but they share no parameters. Layers outside $\mathcal{S}$ are left untouched. Our decoder is the frozen base model with these adapters installed : $p_\theta(u|c, z) = \prod_{t=1}^{T} p_\theta(u^t|u^{<t}, c, z)$. The adapter parameters θ

are the only trainable parameters, and z enters the network solely through T_l^*. Figure 1 illustrates how the steering mechanism is installed.

Encoder We wish to take advantage of the rich structure already present in the base model's backbone, and to avoid training a whole new encoder from scratch in our setup. Therefore, we define our encoder as a shallow MLP on top of frozen backbone's final representation $s_i = \mathrm{pool}(h_L(c_i, u_i))$, with pool implemented as mean pooling of the (user only) utterance tokens' embeddings at the final layer. We denote $\mathcal{N}(\mu_\phi, \mathrm{diag}(\sigma_\phi))$ the posterior distribution, and l_ϕ the logits for the cluster probabilities, both predicted by the encoder:

$$(\mu_\phi, \log \sigma_\phi^2, l_\phi) = \mathrm{MLP}_\phi(s_i), \quad q_\phi(z|c_i, u_i) = \mathcal{N}(\mu_\phi, \mathrm{diag}(\sigma_\phi^2)),$$
$$\tilde{\pi}_k = q_\phi(y = k|c_i, u_i) = \mathrm{softmax}(l_\phi)$$

Unlike VaDE, we amortize the cluster posterior with a dedicated head $\tilde{\pi}$ rather than computing it in closed form from the mixture.

Priors We implement learned priors, in which the mixture parameters $\{\pi_k, \mu_k, \sigma_k\}_{k=1}^K$ are all updated by backpropagation. The weights are stored as logits and mapped to probabilities by a softmax : no uniformity assumption, which would be unreasonable for User Simulation (some behaviors are more likely than others), is imposed on the mixture, allowing training to start at empirical cluster proportions. Initialization follows VaDe's recipe : we pretrain the encoder as a plain autoencoder, fit a Gaussian Mixture to the dataset projected into latent space by the pretrained encoder, and initialize all mixture parameters from the fitted components. We choose to let K be computed by hyperparameter tuning before the start of the training phase. More details are available in Appendix J.

Training Objectives During pretraining, we train parameters μ_ϕ, σ_ϕ and θ without the KL terms (under only token cross-entropy, z is still sampled from the posterior), with VICReg-style (Bardes et al., 2021) variance and norm hinges that prevent dimensional collapse and bound the latent scale. During training we minimize VaDe's negative ELBO, which exercises all heads of the encoder, with cyclical β-annealing and free bits on KL terms. VaDe is prone to cluster collapse when the cluster weights are learned parameters, as the model is free to attribute zero probability to all but one cluster, effectively converting itself into a vanilla VAE. We therefore follow IMSAT (Hu et al., 2017)'s line of work and reward batch entropy $\mathcal{H}(\cdot)$ and penalize per sample entropies of the posterior's assignments $\tilde{\pi} = q_\phi(y|c, u)$. More details about the exact training objective are available in the Appendix A.2. Let B be a batch of samples and $\lambda_{\mathrm{bal}}, \lambda_{\mathrm{conf}}$ hyperparameters. We denote $\bar{\tilde{\pi}} = \frac{1}{|B|}\sum_i \tilde{\pi}_i$ the batch marginal probabilities.

$$\mathcal{L}_{\mathrm{train}} = \mathbb{E}_B[\mathcal{L}_{\mathrm{nELBO}}] - \lambda_{\mathrm{bal}}\mathcal{H}(\bar{\tilde{\pi}}) + \lambda_{\mathrm{conf}}\frac{1}{|B|}\sum_i \mathcal{H}(\tilde{\pi}_i)$$

3.2 Norm-Bounded Latent Steering for Frozen Language Models

This section we describe how each adapter layer T_l^* computes its steering vector to be applied to the model activations. All norms in this paper are euclidean, and computed at the token level. We first apply the block's original frozen normalization : $X_l = \mathrm{LN}_l^{\mathrm{base}}(h_l)$. The whole intervention can be summarized as transform M_l and shift b_l operations computed from z by shallow Multi-layer Perceptrons $a_{\theta,l}$. One distinct adapter layer $a_{\theta,l}$ is installed at each install site l, with no parameter sharing. The normalization of z, performed here via a learned layer normalization layer LN_θ, is crucial for the stability of the algorithm : during early training, the variance on z is very significant due to the cyclical annealing schedule employed (so the posterior has an unconstrained scale at early training), leading to exploding gradients.

The steering mechanism is applied only at utterance token positions. Context positions pass through the frozen normalization unchanged, at both training and inference. For better readability, we have omitted the masking mechanism from the presented expressions. In order to counteract overfitting phenomenons observed during our early experiments, we impose a bound to the perturbation at each install site using a learned gate α_l and a maximum strength hyperparameter τ. $\alpha_l \in [-1, 1]^d$ is an element wise scaling factor of the normalized steering vector $\bar{\delta}_l$ and is computed by $g_{\theta,l}$:

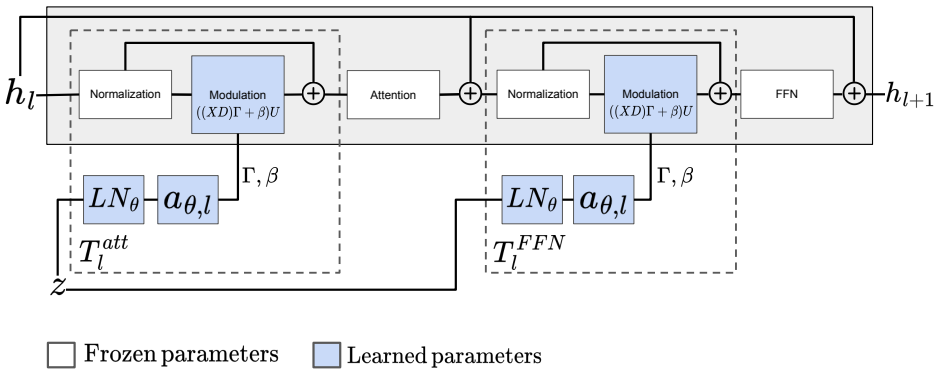


Figure 1: Architecture and installation of the steering adapters T_l^{att} and T_l^{ffn} at a given layer $l \in \mathcal{S}$. We illustrate how they are installed in a typical Transformer decoder block. For this illustration, we show the Low Rank steering variant of the steering mechanism and omit the gating mechanism for better readability.

$\mathbb{R}^r \to [-1,1]^d$, a shallow multi-layer perceptron with **tanh** activation. We denote ΔX_l the steered activations at a given install site l. In order to reduce the memory requirements of $g_{\theta,l}$, the gate reads a low-dimensional summary of the activations: computed by a learned per-layer projection matrix $W_l \in \mathbb{R}^{d\times p}$, $p \ll d$ ($p = 32$ in all experiments).

$$\delta_l = X_l M_l + b_l, \quad \bar{\delta}_l = \delta_l / \|\delta_l\| \tag{1}$$

$$\alpha_l = g_{\theta,l}\big(X_l W_l\big), \quad \Delta X_l = X_l + \tau \|X_l\| (\alpha_l \odot \bar{\delta}_l) \tag{2}$$

We consider two variants of M_l and b_l, detailed below.

Full dimensional Following AdaLN-zero (Peebles & Xie, 2023) and LinACT (Rodriguez et al., 2024), we use a per-channel modulation, where each dimension is rescaled and shifted independently following an affine transformation, with no mixing across channels.

$$a_{\theta,l}^{\text{affine}}(\text{LN}_\theta(z)) = [\gamma_l; \beta_l] \in \mathbb{R}^{2d}, \quad M_l = \text{diag}(\gamma_l), \quad b_l = \beta_l \tag{3}$$

$$\delta_l = X_l \odot \gamma_l + \beta_l \tag{4}$$

Low-rank Given the low dimensionality of the latent variable z carrying the persona information, we take inspiration from LoRA (Hu et al., 2021) and ReFT (Wu et al., 2024), and formulate the hypothesis that only low-rank transformations are necessary in order to compute an appropriate δ_l. Additionally, it becomes only slightly more costly to let $X_l M_l$ become a full linear transformation in lower dimension, while also being more expressive than an equivalent Hadamard product at the same dimensionality. We introduce a per-layer down-projection $D_l \in \mathbb{R}^{d\times r}$ and up-projection $U_l \in \mathbb{R}^{r\times d}$ matrices. Both are z-independent, and not shared with the gate's W_l, so that the rank r of the map and the gate width p remain independent choices. During experiments, we use $r = 16$. The entire conditioning is done in low-dimensions :

$$a_{\theta,l}^{\text{proj}}(LN_\theta(z)) = [G_l; \beta_l] \in \mathbb{R}^{r^2+r}, \quad \Gamma_l = \text{reshape}(G_l) \in \mathbb{R}^{r\times r}, \quad \beta_l \in \mathbb{R}^r \tag{5}$$

$$M_l = D_l \Gamma_l U_l \quad b_l = \beta_l U_l \quad \delta_l = ((X_l D_l)\Gamma_l + \beta_l) U_l \tag{6}$$

Norm-bounded guarantees By construction, our resulting gating mechanism ensures that the norm of the original activations stay bounded relative to the original norm :

$$(1-\tau)\|X_l\| \le \|\Delta X_l\| \le (1+\tau)\|X_l\|$$

Proof is provided in Annex B. The steered activations remain within a relative shell of width 2τ around each token's original norm. This design choice, aimed to allow more freedom to the steering adapters, is therefore a relaxation of norm-preserving activation steering methods like GeoSteer (Ngo et al., 2026), which keep activations within the sphere defined by the original norm. We ablate the effects of τ in our experiments.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We train on an English, non-toxic, non-redacted subset of WildChat-1M (Zhao et al., 2023), restricted to `gpt-3.5-turbo` conversations, with at least two user turns and no turn longer than 1,000 characters to remove pasted content. Of 344,940 qualifying conversations, 85,494 survive this filtering; user-hashed splits yield 4,000 training, 2,000 held-out, and a disjoint evaluation pool. We steer Qwen3.5-4B (Qwen Team, 2026), using its existing chat template to predict user utterances as assistant turns. The system prompt is minimal so behavior is carried by z; prompts are in Appendix C.

Qwen3.5-4B has 32 decoder layers with full attention at layers $3, 7, \ldots, 31$. We install two adapters per full-attention layer from l_0 onward: $l_0 = 15$ gives 5 steered layers and $l_0 = 7$ gives 7.

Training details We use $d_z = 32$ and $K = 8$ components, with K chosen before training by a persona-agreement probe (Appendix J). After 2 epochs of pretraining, we train for 30 epochs with three 10-epoch KL annealing cycles (Fu et al., 2019), β_{KL} rising linearly from 0 to 0.2 over the first 40% of each, and free bits (Kingma et al., 2016) of 0.15 nats per dimension. Batches hold 32,000 tokens. Optimizer settings are in Appendix D. A run takes $\sim$10 h ($l_0 = 15$) or $\sim$15 h ($l_0 = 7$) on a single 48 GB GPU, where l_0 is the first install site. At inference, z is drawn from the prior and we sample with temperature 0.9 and top-p 0.95. Trainable parameters total 54.5 M (full-dimensional, $l_0 = 15$), 76.3 M (full-dimensional, $l_0 = 7$) and 7.9 M (low-rank, $l_0 = 7$, $r = 16$), against a frozen 4B backbone. Epochs and configurations are selected by a rule fixed before the ablation grid was trained: an in-training register signal on held-out conversations, subject to the CE constraint (Section 4.4; details in Appendix H). This signal is used for selection only, and every number we report is recomputed on new prior-sampled generations, from evaluation-pool users, under scorers independent of selection.

4.2 BASELINES

No system receives information about a real user, and each generates every user turn itself. MultiUS draws z from the prior. Our main baseline is UserLM-8B (Naous et al., 2025), fully fine-tuned on WildChat to play the user and among the strongest available trained simulators. Although it is trained to condition on an user intent summarizing the target conversation, we run it *without* an intent as it is the condition that matches ours (the user is not known in advance) and that is also reported by the authors (Naous et al., 2025, Table 1). We use the authors' template and sampling (temperature 1.0, top-p 0.8) with a 1,024-token turn limit, both alone and with guardrails 1 (first-word filter) and 4 (repeat resampling) of their Appendix C.1. We omit guardrails 2 (forbidding the end of the conversation) and 3 (capping turn length), which would fix by construction two quantities we measure: when and how long users write. We further prompt Qwen3.5-4B, our backbone, and GPT-5.6-Luna, a frontier assistant, to play a human user, with a prompt (Appendix C) that asks for imperfect human writing, reuses the UserLM authors' role-play instructions for prompted baselines, and asks the model to silently choose a persona and goal, the prompt analogue of sampling z.

4.3 EVALUATION PROTOCOL

We report results on held-out WildChat samples, as well as on PRISM (Kirk et al., 2024), a diverse human feedback dataset for preference and value alignment of LLMs. Each tested simulator is used to generate 1,500 conversations of at most 8 user turns with the same live assistant (GPT-5.6-Luna, no reasoning, at most 1,000 tokens per reply). One seed is generated and shared across simulators for each conversation, and paired with 1,500 conversations drawn from the evaluation sets (held-out

Table 1: Adapter family, depth l_0 and maximum steering strength τ. Each configuration writes 3,000 first messages from prior samples. The human reference is the real first messages of 3,000 evaluation-pool conversations (human-vs-human control: -0.05 [-0.17, $+0.07$] Mistral, -0.10 [-0.25, $+0.03$] Qwen). Margin: unsteered minus steered held-out CE (unsteered CE $= 3.161$), 60 conversations, paired bootstrap. Verdict: FAIL if the CI lies below 0. $\tau = 1.0$ fails at both depths with the full-dimensional adapters and was therefore not run for the low-rank family. †20 epochs instead of 30.

				$\Delta \log \text{ppl (median)} \to 0$				
adapters	τ	l_0	params	Mistral	Qwen	MAUVE ↑	CE margin [95% CI]	verdict
full-dim.	0.3	15	54.5 M	-0.88 [-0.97, -0.78]	-1.72	0.624 [0.595, 0.652]	$+0.356$ [$+0.236$, $+0.485$]	pass
full-dim.	0.6	15	54.5 M	-0.80 [-0.89, -0.71]	-1.48	0.617 [0.582, 0.644]	$+0.031$ [-0.089, $+0.160$]	tie
full-dim.	1.0	15	54.5 M	-0.69 [-0.79, -0.58]	-1.31	0.638 [0.609, 0.675]	-0.257 [-0.390, -0.117]	FAIL
full-dim.	0.3†	7	76.3 M	-0.89 [-0.98, -0.80]	-1.70	0.598 [0.573, 0.631]	$+0.377$ [$+0.264$, $+0.497$]	pass
full-dim.	**0.6**	**7**	76.3 M	-0.80 [-0.89, -0.69]	-1.46	0.595 [0.561, 0.618]	$+0.008$ [-0.125, $+0.152$]	tie
full-dim.	1.0	7	76.3 M	-0.72 [-0.82, -0.63]	-1.41	0.624 [0.604, 0.650]	-0.169 [-0.306, -0.021]	FAIL
low-rank	0.3†	7	7.9 M	-0.97 [-1.05, -0.87]	-1.77	0.597 [0.561, 0.628]	$+0.283$ [$+0.167$, $+0.406$]	pass
low-rank	0.6	7	7.9 M	-0.66 [-0.75, -0.56]	-1.26	0.578 [0.552, 0.602]	-0.222 [-0.355, -0.075]	FAIL

Wildchat or PRISM). Ending the conversation is part of the simulated behavior, except at the first turn, where it is resampled, since every real conversation has at least one user turn. We report the results for only the first user message and for all user turns.

4.4 METRICS

MAUVE (Pillutla et al., 2021) compares the distribution of generated and human texts in a feature space, and penalizes both unrealistic text and missing modes. We use GPT-2-large features of texts truncated to 256 tokens, $n/10$ quantization buckets and the same n for every system of a table. MAUVE is sensitive to k-means seeding at this sample size, so the point estimate is the mean of 5 runs, and 95% confidence intervals come from subsampling 80% of clusters (conversations, or users for PRISM and unfiltered WildChat) without replacement.

Register ($\Delta \log \text{ppl}$) measures whether the simulator writes as fluently as humans do. For each user message, we compute its log-perplexity (mean token negative log-likelihood) as plain text, without prompt or chat template, under a scorer language model. The metric is the median over a system's messages minus the median over the human reference. Therefore, negative values mean more fluent than humans, which is the characteristic failure of assistant models playing users. We report the median, because real-message log-perplexity is heavy-tailed. We use two scorers from different families: Mistral-7B-v0.3 (Jiang et al., 2023) (neutral) and Qwen3.5-4B (MultiUS's own base model). The 95% CIs are computed from 2,000-resample bootstraps over clusters, system and reference resampled independently.

CE constraint This constraint is based on 60 seeded held-out conversations. We compute the token cross-entropy (CE) of the real user turns with steering on (with z the posterior mean of that conversation) and off. The margin is the unsteered minus the steered CE, bootstrapped over conversations. A configuration fails if the 95% CI of its margin lies below zero: its steering then makes held-out users *less* predictable than no steering at all, meaning it has fit the training users rather than the population. Since z is inferred from the very conversation being scored, this is a lenient bar, and failing it is strong evidence of overfitting.

Behavioral statistics are computed from the generated conversations. We report the number of user turns, probability of a single-turn or a maximum-length (8-turn) conversation, the median message length, the first-turn diversity ($100 \times (1-$ mean pairwise word-set Jaccard similarity) over first messages, as in Naous et al. (2025, Table 2)), echo (a user turn copying the previous assistant reply) and self-repetition (a user turn identical to an earlier turn of the same user). We also report the percentage of utterances generated that are identical to utterances in the training set.

Table 2: Generations against a live assistant (n = 1,500). MAUVE (↑) against three human references, and register $\Delta \log \text{ppl}$ (median). *Ceiling*: other real users of the same corpus. *Shift*: real filtered-WildChat users scored against that corpus. Every MultiUS model beats every baseline on every MAUVE column (paired subsampling test) and is closer to human register on every register column (non-overlapping 95% CIs). All 95% CIs, the Qwen scorer and the PRISM unguided subset are in Appendix E.

	MAUVE ↑						$\Delta \log \text{ppl (Mistral)} \to 0$			
	first message			all turns			first message		all turns	
system	WildChat	PRISM	unfilt.	WildChat	PRISM	unfilt.	WildChat	PRISM	WildChat	PRISM
ceiling	0.964	0.956	0.892	0.895	0.948	0.844	+0.12	−0.02	+0.09	0.00
shift	—	0.112	0.263	—	0.102	0.437	—	+0.71	—	+0.56
MultiUS full-dim., τ=0.6	**0.676**	**0.105**	**0.217**	**0.573**	0.095	0.337	−0.77	−0.06	**−0.54**	+0.02
MultiUS low-rank, τ=0.3	0.615	0.097	0.206	0.522	**0.108**	**0.343**	−0.95	−0.24	−0.82	−0.26
MultiUS low-rank, τ=0.6[‡]	0.608	0.092	0.186	0.495	0.089	0.309	**−0.72**	**−0.01**	−0.55	**+0.01**
UserLM-8b	0.066	0.012	0.061	0.044	0.011	0.059	−3.41	−2.70	−3.49	−2.93
UserLM-8b + guardrails[§]	0.056	0.011	0.057	0.048	0.010	0.058	−3.39	−2.68	−3.42	−2.86
Qwen3.5-4B prompted	0.008	0.013	0.007	0.008	0.012	0.009	−1.71	−1.00	−1.44	−0.88
GPT-5.6-luna prompted	0.004	0.004	0.004	0.005	0.006	0.006	−1.86	−1.16	−1.01	−0.45
no steering (ablation)	0.006	0.006	0.005	0.011	0.009	0.019	−1.49	−0.78	−1.68	−1.12

[‡]Fails the CE constraint (Table 1). [§]300-token turn limit; without guardrails, the 300- and 1,024-token limits give the same MAUVE (0.065 vs. 0.066, first message vs. WildChat). *unfilt.*: unfiltered WildChat.

4.5 Ablations

Steering strength τ and steering depth l_0 interaction We compare different combinations of τ and l_0. Register moves monotonically with τ at both depths and for both adapter families: stronger steering favors human-likeness. However, the CE constraint moves in the opposite direction: $\tau = 0.3$ makes held-out users significantly more predictable, $\tau = 0.6$ is neutral, and $\tau = 1.0$ makes them significantly less predictable at both depths. We conclude the register gained by steering harder is bought by overfitting the training users. Steering from layer 7 instead of 15 changes register by at most 0.03 nats under Mistral (0.10 under Qwen, at $\tau = 1.0$). The selection signal favored depth (Appendix I), which is why the selected configuration is $l_0 = 7$. MAUVE on first messages is flat across all configurations (0.58–0.64): every configuration that trains is far closer to humans than any baseline (Table 2), and MAUVE does not separate them.

Parameter efficiency The low-rank variant, which operates entirely in an $r = 16$ dimensional subspace, use 9.65× fewer parameters than the full-dimensional one at the same depth. At $\tau = 0.3$ they pass the CE constraint, match the full-dimensional adapters on MAUVE (0.597 vs. 0.598), and stay within 0.08 nats of them on register. At $\tau = 0.6$ they reach the closest register of any configuration but fail the CE constraint: at equal τ, a low-rank map makes the adapters *stronger*, not weaker, and the admissible τ is lower. In multi-turn conversations, the low-rank $\tau = 0.3$ model reaches 91% of the full-dimensional model's MAUVE against WildChat and 92% against PRISM (Table 2).

4.6 Results

Human-likeness MultiUS obtains the highest MAUVE against every reference (Table 2), on first messages, which no assistant reply can influence, as on all user turns. Against held-out WildChat users it reaches 0.676 (full-dimensional) and 0.615 (low-rank), against 0.066 for UserLM-8B and at most 0.008 for prompted simulators. The most interesting results are obtained on PRISM, a population and set of assistants no system was trained on. MultiUS scores 0.105, statistically indistinguishable from real WildChat users scored against the same corpus (0.112): it matches the cross-population reference, the best a simulator trained on WildChat can be expected to achieve. The ordering also holds on unfiltered WildChat, whose distribution is closer to UserLM's training data (0.217 vs. 0.061), where MultiUS remains below real users (0.263) since it cannot produce the pasted content excluded from our training data. MAUVE is sensitive to length, and the baselines write longer messages than humans. If we restrict every set to messages within the human 10th–90th

Table 3: Conversation-level behavior (n = 1,500). The human row is the same 1,500 evaluation-pool WildChat conversations, filtered like the training data ($P(1 \text{ turn}) = 0$). Training copies: share of user turns of at least 20 characters that are identical, after normalization, to an user turn of our training conversations.

system	user turns (mean)	$P(1 \text{ turn})$	$P(8 \text{ turns})$	first-turn chars (median)	all-turn chars (median)	first-turn diversity	echo	self-repeat	training copies (%)
human (WildChat)	3.66	0.000	0.103	67	64	97.5	0.004	0.026	0.08
MultiUS, full-dim. (τ=0.6)	3.90	0.004	0.131	90	70	95.6	0.004	0.079	1.82
MultiUS, low-rank (τ=0.3)	3.65	0.013	0.119	90	76	95.0	0.004	0.098	0.58
MultiUS, low-rank (τ=0.6)	3.88	0.001	0.138	90	77	95.6	0.003	0.083	3.54
UserLM-8b, no intent	3.34	0.571	0.293	3161	2558	96.7	0.007	0.458	0.00
UserLM-8b, no intent, guardrails 1,4[§]	3.00	0.621	0.252	1012	874	97.4	0.006	0.256	0.00
Qwen3.5-4B, prompted	7.38	0.000	0.719	138	193	87.5	0.000	0.001	0.03
GPT-5.6-luna, prompted	5.20	0.000	0.087	144	155	68.4	0.000	0.000	0.00
ablation: no steering	7.91	0.004	0.970	70	458	84.5	0.000	0.001	0.00

[§]300-token turn limit.

length percentile, on all user turns (UserLM has too few human-length first messages), UserLM's MAUVE against WildChat rises from 0.068 to 0.188 but stays well below MultiUS (0.782). While 99% of GPT-5.6-Luna messages already have a human length, its MAUVE stays at 0.006 (Appendix G).

Register All simulators are more fluent than real users ($\Delta \log \text{ppl} < 0$. Under the neutral Mistral scorer, every MultiUS configuration is closer to human register than every baseline, and both $\tau = 0.6$ models are statistically indistinguishable from real PRISM users (-0.06 [-0.14, 0.02] and -0.01 [-0.10, 0.07]). Under the Qwen scorer, from the backbone's own family, the same holds for the $\tau = 0.6$ models, while the low-rank $\tau = 0.3$ model is on par with prompted simulators (Appendix E). With steering switched off, the base model is 0.7 nats more fluent, and prompted simulators, although instructed to write like imperfect humans, remain 1.0–1.9 nats too fluent, therefore we can conclude that steering effectively improves register. UserLM's very low $\Delta \log \text{ppl}$ (-3.41) reflects degenerate text rather than a chatbot register: 59% of its first turns reach the 1,024-token limit, against 4.4% for real users, most of them repetition loops (Appendix F).

Conversational behavior. MultiUS reproduces the shape of human conversations (Table 3): 3.9 user turns against 3.7, and 13% of conversations reaching the 8-turn limit against 10%. Without steering, the same model rarely stops (97% reach the limit), and prompted simulators also stop late (72% for Qwen), with GPT-5.6-Luna's first messages particularly homogeneous (diversity 68.4 against 97.5). UserLM ends 57% of conversations after one turn, matching the unfiltered WildChat population it was trained on (68.5%) rather than the multi-turn population we model. MultiUS's main defects are self-repetition, three times the human rate (0.079 against 0.026), and verbatim copies of training prompts in 1.8% of turns. This effect is reduced in the low-rank $\tau = 0.3$ model.

4.7 Limitations

As presented in this work, MultiUS does not model users who paste long content, and repeats itself three times more often than humans. Moreover, its mixture components are discovered without supervision and are not guaranteed to be interpretable, and the latent does not explicitly separate user intent from user profile, which may or me not matter depending on the use case.

5 Conclusion

We introduced MultiUS, a cheap and effective user simulator that discovers a mixture of user behaviors without supervision and realizes each through norm-bounded activation steering. Adding under 2% of trainable parameters to a 4B model, MultiUS's behavior is far closer to real users' than a fully fine-tuned 8B user model and prompted simulators, and matches the cross-population reference on PRISM. Our ablations reveal the steering bound τ trades off more human register against overfitting to training users, and show that low-rank adapters remain competitive with far fewer parameters. MultiUS is a step toward activation steering without behavior demonstrations, and we believe finer control over user intent and interpretable profiles are natural next steps.

AI USE STATEMENT

Literature Review After doing our own literature review using conventional methods (Semantic Scholar, Google searches, reading related work sections of other relevant papers, etc.), we have used generative AI tools as a final safeguard to identify any relevant literature that we might have missed. Generative AI also helped us understand some methods more in depth by analyzing the released open-source implementations.

Writing We used generative AI tools to help with spotting spelling and grammatical mistakes, improve writing fluency, as well as to help spot problems in our math expressions, like variable name collisions. AI also helped format tables correctly into LaTeX format. All text was written by the authors. AI was used solely to correct errors and improve the fluency of the text. We also tried feeding sections of paper to an AI tool, with a simple prompt asking if it understood all the work that was done : this was a "test" of how easy to understand our writing was. Finally, AI helped us brainstorm ideas for choosing the paper's title.

Coding The main implementation of MultiUS was written by hand by the authors. However, we used AI coding tools for :

1. Solving memory efficiency bugs that were making training on longer conversations impossible.
2. Refactoring the code for better readability and usability.
3. Making quick changes during the developing phase to try new ideas faster (for example, tweaking hidden layer sizes, hyperparameters, etc.).
4. Helping create overnight training and evaluation scripts.
5. Analyzing the code for any lines that risked exposing the identity of the authors.

A particularly interesting use case we have found for AI tools was having an agent monitor experiments overnight : an agent regularly monitored experiments, and was instructed to restart training in case of crashes, shutdown the machine when experiments finished earlier than expected and provide updates to the authors via messaging applications.

We did not use generative AI to generate figures, proving mathematical claims, writing proofs, interpreting results, processing datasets or discovering research topics. Synthetic data generation, translation and formulating questions for surveys/interviews are not applicable to this work

We have reviewed all AI-assisted work in depth by reading and testing the code for correctness (2 authors were involved in this process). We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

In this work, we steer the Qwen3.5-4B model, a post-trained model. However, our activation steering method carries the risk of weakening the model's guardrails, increasing the risk of the model generating toxic and otherwise unsafe utterances. Moreover, MultiUS improves the human-likeness of the steered model's utterances, making it more difficult for humans and AI detection systems to tell them apart from a real human's. We have developed MultiUS for beneficial purposes such as LLM evaluation and training, and believe that it is unlikely that the method could be applied for malicious purposes. However, with the rise of more and more human-like User Simulators, we hope our work helps encourage future research to differentiate User Simulator utterances from real humans utterances.

REPRODUCIBILITY STATEMENT

We release, as anonymized supplementary material, the code for data construction, training, generation and evaluation in this paper, together with the configuration of each reported run. The data construction is fully specified in Section 4.1: WildChat-1M and PRISM are public, and our filters, user-disjoint splits and all sampling seeds are fixed in the released scripts, so the exact training,

held-out and evaluation sets can be rebuilt from the public data. The model and adapter architecture are given in Section 3, the exact training objective in Appendix A.2, all hyperparameters and the checkpoint-selection rule in Section 4.1, and all prompts in Appendix C. The evaluation protocol, references and metrics, including MAUVE's settings and the bootstrap procedures, are described in Sections 4.3 and 4.4. Each training run fits on a single 48 GB GPU (10–15 h). The GPT-5.6-Luna assistant LLM is accessed through a proprietary API, which may change over time; we therefore also release the trained adapter weights of the reported models and the 1,500 generated conversations of every system, from which every number in Tables 2–1 and in the appendix can be recomputed without API access.

REFERENCES

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik R. Narasimhan. $\tauˆ2$-Bench: Evaluating Conversational Agents in a Dual-Control Environment. In *Forty-third International Conference on Machine Learning*, June 2026. URL https://openreview.net/forum?id=OC2z7iSQKa.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating Sentences from a Continuous Space. In Stefan Riezler and Yoav Goldberg (eds.), *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002/.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona Vectors: Monitoring and Controlling Character Traits in Language Models, September 2025. URL http://arxiv.org/abs/2507.21509. arXiv:2507.21509 [cs.CL].

Harshita Chopra, Kshitish Ghate, Aylin Caliskan, Tadayoshi Kohno, Chirag Shah, and Natasha Jaques. Beyond Cooperative Simulators: Generating Realistic User Personas for Robust Evaluation of LLM Agents. In *Second Workshop on Social Simulation with LLMS: Fidelity in Applications*, August 2026. URL https://openreview.net/forum?id=cJVWKXCws0.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders, January 2017. URL http://arxiv.org/abs/1611.02648. arXiv:1611.02648 [cs.LG].

Alex Ferrando, Xavier Suau, Jordi Gonzàlez, and Pau Rodriguez. Dynamically Scaled Activation Steering. *Transactions on Machine Learning Research*, June 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=EXKv1KMNtR.

Rafael Ferreira, David Semedo, and Joao Magalhaes. Multi-trait User Simulation with Adaptive Decoding for Conversational Task Assistants. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16105–16130, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.945. URL https://aclanthology.org/2024.findings-emnlp.945/.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 240–250, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1021. URL https://aclanthology.org/N19-1021/.

Muyu He, Anand Kumar, Soumyadeep Bakshi, James Zou, and Nazneen Rajani. Impatient Users Confuse AI Agents: High-fidelity Simulations of Human Traits for Testing Agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37569–37602, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1743. URL https://aclanthology.org/2026.acl-long.1743/.

Anna Hedström, Salim I. Amoukou, Tom Bewley, Saumitra Mishra, and Manuela Veloso. To Steer or Not to Steer? Mechanistic Error Reduction with Abstention for Language Models. In *Forty-second International Conference on Machine Learning*, June 2025. URL https://openreview.net/forum?id=fUCPq5RvmH.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, February 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, October 2021. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning Discrete Representations via Information Maximizing Self-Augmented Training. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1558–1567. PMLR, July 2017. URL https://proceedings.mlr.press/v70/hu17b.html.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs.CL].

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 1965–1972, Melbourne, Australia, August 2017. AAAI Press. ISBN 978-0-9992411-0-3.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 4743–4751, Red Hook, NY, USA, December 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9. URL https://dl.acm.org/doi/10.5555/3157382.3157627.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM Alignment Dataset: What Participatory, Representative and Individualised Human Feedback Reveals About the Subjective and Multicultural Alignment of Large Language Models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, November 2024. URL https://openreview.net/forum?id=DFr5hteojx.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs Get Lost In Multi-Turn Conversation. In *The Fourteenth International Conference on Learning Representations*, October 2025. URL https://openreview.net/forum?id=VKGTGGcwl6.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming Refusal with Conditional Activation Steering. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=Oi47wc10sm.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4678–4699, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.378. URL https://aclanthology.org/2020.emnlp-main.378/.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: eliciting truthful answers from a language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pp. 41451–41530, Red Hook, NY, USA, December 2023. Curran Associates Inc.

Yajiao Liu, Xin Jiang, Yichun Yin, Yasheng Wang, Fei Mi, Qun Liu, Xiang Wan, and Benyou Wang. One Cannot Stand for Everyone! Leveraging Multiple User Simulators to train Task-oriented Dialogue Systems. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1–21, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.1. URL https://aclanthology.org/2023.acl-long.1/.

Shuhaib Mehri, Philippe Laban, Sumuk Shashidhar, Marwa Abdulhai, Sergey Levine, Michel Galley, and Dilek Hakkani-Tür. Measuring and Mitigating the Distributional Gap Between Real and Simulated User Behaviors, May 2026. URL http://arxiv.org/abs/2605.07847. arXiv:2605.07847 [cs.CL].

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the Dialogue: Training and Evaluating User Language Models. In *The Fourteenth International Conference on Learning Representations*, October 2025. URL https://openreview.net/forum?id=ykSmkVqzn4.

Xuan Cuong Ngo, Hao Vo, and Ngan Le. GEOSTEER: Geodesic Optimization for Activation Steering in Large Language Models, September 2026. URL http://arxiv.org/abs/2609.10658. arXiv:2609.10658 [cs.LG].

Davide Paglieri, Logan Cross, William A. Cunningham, Joel Z. Leibo, and Alexander Sasha Vezhnevets. Persona Generators: Generating Diverse Synthetic Personas for Arbitrary Contexts, May 2026. URL http://arxiv.org/abs/2602.03545. arXiv:2602.03545 [cs.AI].

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers. In *Advances in Neural Information Processing Systems*, November 2021. URL https://openreview.net/forum?id=Tqx7nJp7PR.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via Contrastive Activation Addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling Language and Diffusion Models by Transporting Activations. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=l2zFn6TIQi.

Pau Rodriguez, Michal Klein, Eleonora Gualdoni, Valentino Maiorca, Arno Blaas, Luca Zappella, Marco Cuturi, and Xavier Suau. LinEAS: End-to-end Learning of Activation Steering with a Distributional Loss. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025. URL https://openreview.net/forum?id=EBONa3tT3K&noteId=4Hydc1JdyZ.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in Simulation: LLM-Simulated Users are Unreliable Proxies for Human Users in Agentic Evaluations. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47423–47439, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2192. URL https://aclanthology.org/2026.acl-long.2192/.

Leheng Sheng, Changshuo Shen, Weixiang Zhao, Junfeng Fang, Xiaohao Liu, Zhenkai Liang, Xiang Wang, An Zhang, and Tat-Seng Chua. AlphaSteer: Learning Refusal Steering with Principled Null-Space Constraint. In *The Fourteenth International Conference on Learning Representations*, October 2025. URL https://openreview.net/forum?id=1vvbzAqdTe.

Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik R. Narasimhan, and Victor Barres. τ-Knowledge: Evaluating Conversational Agents over Unstructured Knowledge. In *Forty-third International Conference on Machine Learning*, June 2026. URL https://openreview.net/forum?id=XHZK5abtw2.

Wenxian Shi, Hao Zhou, Ning Miao, and Lei Li. Dispersed Exponential Family Mixture VAEs for Interpretable Text Generation. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8840–8851. PMLR, November 2020. URL https://proceedings.mlr.press/v119/shi20f.html.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. Parrot: Enhancing Multi-Turn Instruction Following for Large Language Models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9729–9750, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.525. URL https://aclanthology.org/2024.acl-long.525/.

Magamed Taimeskhanov, Samuel Vaiter, and Damien Garreau. Towards Understanding Steering Strength. In *Forty-third International Conference on Machine Learning*, June 2026. URL https://openreview.net/forum?id=A52LNgpGKy.

Kuang Wang, Xianfei Li, Shenghao Yang, Li Zhou, Feng Jiang, and Haizhou Li. Know You First and Be You Better: Modeling Human-Like User Simulators via Implicit Profiles. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21082–21107, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1025. URL https://aclanthology.org/2025.acl-long.1025/.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation Finetuning for Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024. URL https://openreview.net/forum?id=fykjplMc0V&noteId=ffrNr0KmhD.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. {τ}-bench: A Benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*, October 2024. URL https://openreview.net/forum?id=roNSXZpUDN.

Zejia You, Chunyuan Deng, and Hanjie Chen. Spherical Steering: Geometry-Aware Activation Rotation for Language Models. In *Forty-third International Conference on Machine Learning*, June 2026. URL https://openreview.net/forum?id=9PPeJsVLdB.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. *Transactions on Machine Learning Research*, November 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=RY19y2RI1O.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *The Twelfth International Conference on Learning Representations*, October 2023. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

A Training Objectives Details

A.1 Pretraining phase

Reconstruction We adopt a simple reconstruction objective, defined as the Cross Entropy to be minimized with respect to the target sequence :

$$\mathcal{L}_{\text{recon}}(u, c; \phi, \theta) = -\sum_{t=1}^{T} \log p_\theta(u^t | u^{<t}, c, z)$$

VICReg Regularization We denote $B = [(u_1, c_1), \ldots, (u_{|B|}, c_{|B|})]$ a batch utterances and contexts, $Z = [(\mu_\phi(u_1, c_1), \ldots, \mu_\phi(u_{|B|}, c_{|B|})] \in \mathbb{R}^{|B| \times d_z}$ the matrix of centroids computed by the encoder from B, and $\sigma_j = \sqrt{\text{Var}(Z_{:,j}) + 10^{-4}}, j \in [1, d_z]$ the computed batch-wise standard deviation. We denote $\mathcal{L}_{\text{std}}^{\min}(B)$ the minimum standard deviation hinge, $\mathcal{L}_{\text{std}}^{\max}(B) = \frac{1}{d_z}\sum_{j=1}^{d_z} \max(0, \sigma_j - v_{max})$ the maximum standard deviation hinge, and $\mathcal{L}_{\text{norm}}(B)$ the maximum norm hinge.

$$\mathcal{L}_{\text{std}}^{\min}(B) = \frac{1}{d_z}\sum_{j=1}^{d_z} \max(0, v_{min} - \sigma_j) \tag{7}$$

$$\mathcal{L}_{\text{std}}^{\max}(B) = \frac{1}{d_z}\sum_{j=1}^{d_z} \max(0, \sigma_j - v_{max}) \tag{8}$$

$$\mathcal{L}_{\text{norm}}(B) = \frac{1}{|B|}\sum_{i=1}^{|B|} \max\left(0, \frac{\|\mu_\phi(c_i, u_i)\|}{\sqrt{d_z}} - n_{max}\right) \tag{9}$$

With v_{min}, v_{max} and n_{max} hyperparameters. The total pretraining loss is simply a weighted sum of the different losses as below :

$$\mathcal{L}_{\text{pretrain}}(B; \phi, \theta) = \mathbb{E}_{u,c\sim B}[\mathcal{L}_{\text{recon}}(u, c; \phi, \theta)] + \lambda_{\text{reg}}(\mathcal{L}_{\text{std}}^{\min}(B) + \mathcal{L}_{\text{std}}^{\max}(B) + \mathcal{L}_{\text{norm}}(B))$$

With λ_{reg} a regularization weighting coefficient. For our experiments, we set $\lambda_{\text{reg}} = 2.0$, $v_{min} = 1.0$, $v_{max} = 3.0$, $n_{max} = 3.0$.

A.2 Training phase

After pretraining is done, we minimize VaDe's negative ELBO as the main training loss, which exercises all heads of the encoder : $\mu_\phi, \log \sigma_\phi^2, q_\phi(y = k|u, c)$.

$$\begin{aligned}\mathcal{L}_{\text{nELBO}}(u, c; \phi, \theta) = \mathbb{E}_{z\sim q_\phi(z|c,u)}[-\sum_{t=1}^{T} \log p_\theta(u^t | u^{<t}, c, z)] \\ +\beta_{KL}(D_{KL}(q_\phi(y|c, u)||p(y)) \\ +\mathbb{E}_{y\sim q_\phi(y|c,u)}[D_{KL}(q_\phi(z|c, u)||p(z|y))])\end{aligned} \tag{10}$$

β_{KL} is a hyperparameter that controls the tradeoff between regularization and reconstruction, following β-VAE's Higgins et al. (2017) approach.

VaDe is prone to cluster collapse when the cluster weights are learned parameters, as the model is free to attribute zero probability to all but one cluster, effectively converting itself into a vanilla VAE. Serving as a guardrail against cluster collapse, we follow IMSAT's (Hu et al., 2017) line of work and reward batch entropy and penalize per sample entropies of the posterior's assignments $\tilde{\pi} = q_\phi(y|c,u)$. We denote $\bar{\tilde{\pi}} = \frac{1}{|B|}\sum_i \tilde{\pi}_i$ the batch marginal probabilities.

$$\mathcal{L}_{\text{train}}(B;\phi,\theta) = \mathbb{E}_{u,c\sim B}[\mathcal{L}_{\text{nELBO}}(u,c;\phi,\theta)] - \lambda_{\text{bal}}\mathcal{H}(\bar{\tilde{\pi}}) + \lambda_{\text{conf}}\frac{1}{|B|}\sum_{i=1}^{|B|}\mathcal{H}(\tilde{\pi}_i)$$

We set $\lambda_{\text{conf}} = 0.05$ and $\lambda_{\text{bal}} = 0.1$ for our experimentation.

B Proof of norm-bounded guarantees

By construction,

$$\begin{aligned}||\Delta X_l|| &\leq ||X_l|| + ||X_l|| \times \tau \times ||\alpha_l \odot \bar{\delta}|| \\ &= (1 + \underbrace{||\alpha_l \odot \bar{\delta}_l|| \times \tau}_{=\epsilon})||X_l||\end{aligned}$$

Additionally, using the inverse triangular inequality :

$$||\Delta X_l|| \geq |||X_l|| - ||X_l|| \times \tau \times ||\alpha \odot \bar{\delta}_l||| = (1-\epsilon)||X||$$

$\bar{\delta}_l$ is normalized, and $|\alpha_{l,i}| \leq 1$. If we consider, $\epsilon = ||\alpha \odot \bar{\delta}_l|| \times \tau \leq \tau$, by construction, our resulting gating mechanism ensures that the norm of the original activations stay within an $\epsilon \leq \tau$ of the original norm :

$$(1-\tau)||X_l|| \leq ||\Delta X_l|| \leq (1+\tau)||X_l||$$

C Prompts

Training and inference on MultiUS uses a prompt that is intentionally minimal, in order to let the behavior information come from the latent variable.

All prompted-baselines use the prompted-baseline system prompt (its paragraph 3 corresponds to the UserLM authors' own role-play prompt for prompted assistant LMs, Naous et al. (2025) Fig. 9. The [INTENT] wildcard in the original prompt is replaced by the model inventing its own persona, the prompt analogue of drawing z from the prior)

C.1 MultiUS prompt

You are the human user in a chat with an AI assistant. Write only your next message to it: no role labels, no surrounding quotation marks, no explanation of what you are doing, and never speak as the assistant. When you are done, or you have lost interest, end the conversation by outputting $<$ END $>$ alone on its own line.

C.2 Prompted-baseline system prompt

You are a human user chatting with an AI assistant, like the millions of people who use chatbots every day. You are not an AI, and you should never write like one.

Before your first message, silently decide who you are and what you want from this conversation. It can be anything a real person might bring to a chatbot. Then write only your next message to the assistant.

Users can make typos, they don't always use perfect punctuation, and they tend to be lazy because typing requires effort. You should also split information across turns and not give everything at the start. However, you should not overdo these things: you must realistically act like a human.

Output only the message itself: no role labels, no quotation marks, no explanation. When your goal is met or you have lost interest, output <END> alone on its own line.

D HYPERPARAMETERS

Component	Optimizer	Learning rate
Encoder	Adam	10^{-3}
Mixture prior	Adam	$5 \cdot 10^{-4}$
Adapters	AdamW (wd 0.01)	$5 \cdot 10^{-4}$
Gates	AdamW (wd 0.01)	$5 \cdot 10^{-3}$

Table 4: Optimizer settings.

E ADDITIONAL RESULTS

Table 5: MAUVE on first user messages, with 95% CIs (n per column: 1,500, 1,500, 1,405).

system	WildChat	PRISM	WildChat unfiltered
ceiling (other real users)	0.964 [0.947, 0.974]	0.956 [0.931, 0.973]	0.892 [0.862, 0.922]
shift (real filtered WildChat users)	—	0.112 [0.093, 0.135]	0.263 [0.237, 0.300]
MultiUS, full-dim. (τ=0.6)	0.676 [0.646, 0.705]	0.105 [0.091, 0.124]	0.217 [0.191, 0.242]
MultiUS, low-rank (τ=0.3)	0.615 [0.574, 0.652]	0.097 [0.083, 0.115]	0.206 [0.179, 0.236]
MultiUS, low-rank (τ=0.6)	0.608 [0.562, 0.648]	0.092 [0.079, 0.105]	0.186 [0.159, 0.215]
UserLM-8b, no intent	0.066 [0.055, 0.074]	0.012 [0.011, 0.013]	0.061 [0.052, 0.071]
UserLM-8b, no intent, 300 tok	0.065 [0.056, 0.075]	0.012 [0.011, 0.014]	0.063 [0.052, 0.073]
UserLM-8b, guardrails 1,4 (300 tok)	0.056 [0.049, 0.065]	0.011 [0.009, 0.012]	0.057 [0.052, 0.063]
Qwen3.5-4B, prompted	0.008 [0.007, 0.009]	0.013 [0.010, 0.015]	0.007 [0.006, 0.008]
GPT-5.6-luna, prompted	0.004 [0.004, 0.004]	0.004 [0.004, 0.004]	0.004 [0.004, 0.004]
ablation: no steering	0.006 [0.005, 0.006]	0.006 [0.005, 0.006]	0.005 [0.005, 0.005]

Table 6: MAUVE on all user turns of the generated conversations (n per column: 4,495, 4,495, 2,470).

system	WildChat	PRISM	WildChat unfiltered
ceiling (other real users)	0.895 [0.876, 0.913]	0.948 [0.930, 0.965]	0.844 [0.810, 0.874]
shift (real filtered WildChat users)	—	0.102 [0.093, 0.113]	0.437 [0.398, 0.484]
MultiUS, full-dim. (τ=0.6)	0.573 [0.532, 0.610]	0.095 [0.084, 0.108]	0.337 [0.306, 0.361]
MultiUS, low-rank (τ=0.3)	0.522 [0.494, 0.557]	0.108 [0.097, 0.124]	0.343 [0.300, 0.378]
MultiUS, low-rank (τ=0.6)	0.495 [0.475, 0.531]	0.089 [0.077, 0.101]	0.309 [0.279, 0.342]
UserLM-8b, no intent	0.044 [0.039, 0.048]	0.011 [0.010, 0.012]	0.059 [0.050, 0.068]
UserLM-8b, no intent, 300 tok	0.045 [0.039, 0.051]	0.012 [0.011, 0.013]	0.063 [0.056, 0.071]
UserLM-8b, guardrails 1,4 (300 tok)	0.048 [0.043, 0.053]	0.010 [0.010, 0.011]	0.058 [0.052, 0.065]
Qwen3.5-4B, prompted	0.008 [0.008, 0.009]	0.012 [0.011, 0.014]	0.009 [0.008, 0.010]
GPT-5.6-luna, prompted	0.005 [0.005, 0.006]	0.006 [0.005, 0.006]	0.006 [0.005, 0.006]
ablation: no steering	0.011 [0.010, 0.013]	0.009 [0.008, 0.010]	0.019 [0.016, 0.023]

Unfiltered WildChat and register. Against the unfiltered reference, the prompted baselines obtain the $\Delta \log \mathrm{ppl}$ closest to zero (e.g. Qwen prompted $+0.07$ [-0.09, $+0.32$] vs. MultiUS $+1.02$ [$+0.85$, $+1.28$], first message, Mistral). This reference mixes two populations with very different perplexity (highly predictable pasted text, and typed text), and a location statistic of a bimodal

Table 7: MAUVE on first user messages against the PRISM *unguided* subset (organic use), $n = 1{,}500$.

system	PRISM unguided
ceiling (other real users)	0.952 [0.937, 0.966]
shift (real filtered WildChat users)	0.141 [0.124, 0.155]
MultiUS, full-dim. (τ=0.6)	0.138 [0.118, 0.154]
MultiUS, low-rank (τ=0.3)	0.131 [0.111, 0.144]
MultiUS, low-rank (τ=0.6)	0.121 [0.104, 0.140]
UserLM-8b, no intent	0.014 [0.012, 0.016]
UserLM-8b, no intent, 300 tok	0.014 [0.012, 0.015]
UserLM-8b, guardrails 1,4 (300 tok)	0.012 [0.011, 0.013]
Qwen3.5-4B, prompted	0.017 [0.014, 0.022]
GPT-5.6-luna, prompted	0.004 [0.004, 0.005]
ablation: no steering	0.007 [0.006, 0.007]

Table 8: Register $\Delta \log \mathrm{ppl}$ (median, 95% CI) under both scorers, first user message.

system	WildChat, Mistral	PRISM, Mistral	WildChat, Qwen	PRISM, Qwen
ceiling (other real users)	$+0.12$ [-0.06, $+0.28$]	-0.02 [-0.09, $+0.05$]	$+0.16$ [-0.05, $+0.34$]	-0.00 [-0.09, $+0.09$]
shift (real filtered WildChat users)	—	$+0.71$ [$+0.58$, $+0.82$]	—	$+0.52$ [$+0.36$, $+0.68$]
MultiUS, full-dim. (τ=0.6)	-0.77 [-0.89, -0.63]	-0.06 [-0.14, $+0.02$]	-1.38 [-1.54, -1.19]	-0.86 [-0.99, -0.74]
MultiUS, low-rank (τ=0.3)	-0.95 [-1.06, -0.82]	-0.24 [-0.31, -0.17]	-1.75 [-1.92, -1.57]	-1.23 [-1.35, -1.11]
MultiUS, low-rank (τ=0.6)	-0.72 [-0.85, -0.58]	-0.01 [-0.10, $+0.07$]	-1.20 [-1.39, -1.02]	-0.68 [-0.81, -0.56]
UserLM-8b, no intent	-3.41 [-3.52, -3.28]	-2.70 [-2.76, -2.64]	-4.57 [-4.71, -4.41]	-4.04 [-4.13, -3.97]
UserLM-8b, no intent, 300 tok	-3.33 [-3.43, -3.20]	-2.62 [-2.68, -2.56]	-4.39 [-4.53, -4.23]	-3.87 [-3.95, -3.79]
UserLM-8b, guardrails 1,4 (300 tok)	-3.39 [-3.49, -3.26]	-2.68 [-2.74, -2.62]	-4.47 [-4.60, -4.30]	-3.94 [-4.02, -3.86]
Qwen3.5-4B, prompted	-1.71 [-1.81, -1.58]	-1.00 [-1.06, -0.94]	-1.84 [-1.98, -1.67]	-1.32 [-1.40, -1.22]
GPT-5.6-luna, prompted	-1.86 [-1.96, -1.74]	-1.16 [-1.21, -1.10]	-1.64 [-1.78, -1.49]	-1.12 [-1.20, -1.04]
ablation: no steering	-1.49 [-1.59, -1.36]	-0.78 [-0.84, -0.73]	-1.52 [-1.67, -1.35]	-1.00 [-1.09, -0.90]

distribution can be matched by a system resembling neither mode. MAUVE, which compares distributions, ranks MultiUS first on this reference (Table 2).

F USERLM TURN LENGTH

Raising the limit from 300 to 1,024 tokens does not produce complete long messages: most capped turns are repetition loops, i.e. the model does not emit its end-of-turn token. This concerns UserLM *without an intent*, at the authors' sampling parameters; we make no claim about its behavior with intents.

G LENGTH CONTROL

Matching each system's length distribution to the human one decile by decile is infeasible for the baselines (e.g. GPT-5.6-luna writes no first message shorter than 39 characters, while 30% of real first messages are). We therefore restrict every set, the reference included, to messages within the reference's 10th–90th character percentile, and score all systems at one common n, the smallest in-window count. Mean $\pm$ sd over 10 draws. MAUVE depends on n: these values are comparable within a table only. First messages are not reported, since UserLM has too few first messages of human length (196 against WildChat, 69 against PRISM).

H CHECKPOINT SELECTION

The selection rule was fixed before the ablation grid was trained. Every epoch is checkpointed and scored on 48 held-out conversations: their first user message is generated twice (up to 40 tokens), once from the posterior mean of that conversation's z and once from a prior sample, and each generation is scored by its log-perplexity under the unsteered base model. The selection signal

Table 9: Register $\Delta \log \mathrm{ppl}$ (median, 95% CI) under both scorers, all user turns. Under Qwen, GPT-5.6-luna's pooled register is closer to human than MultiUS's; under Mistral the order reverses.

system	WildChat, Mistral	PRISM, Mistral	WildChat, Qwen	PRISM, Qwen
ceiling (other real users)	$+0.09$ [-0.03, $+0.19$]	-0.00 [-0.07, $+0.05$]	$+0.08$ [-0.06, $+0.22$]	$+0.00$ [-0.07, $+0.08$]
shift (real filtered WildChat users)	—	$+0.56$ [$+0.46$, $+0.64$]	—	$+0.40$ [$+0.28$, $+0.51$]
MultiUS, full-dim. (τ=0.6)	-0.54 [-0.64, -0.44]	$+0.02$ [-0.06, $+0.09$]	-0.97 [-1.10, -0.83]	-0.57 [-0.66, -0.46]
MultiUS, low-rank (τ=0.3)	-0.82 [-0.92, -0.72]	-0.26 [-0.34, -0.19]	-1.43 [-1.56, -1.30]	-1.04 [-1.14, -0.94]
MultiUS, low-rank (τ=0.6)	-0.55 [-0.63, -0.44]	$+0.01$ [-0.06, $+0.09$]	-0.93 [-1.06, -0.79]	-0.53 [-0.63, -0.42]
UserLM-8b, no intent	-3.49 [-3.58, -3.41]	-2.93 [-2.99, -2.88]	-4.65 [-4.76, -4.52]	-4.25 [-4.33, -4.17]
UserLM-8b, no intent, 300 tok	-3.42 [-3.50, -3.33]	-2.86 [-2.92, -2.81]	-4.48 [-4.59, -4.36]	-4.08 [-4.17, -4.00]
UserLM-8b, guardrails 1,4 (300 tok)	-3.42 [-3.50, -3.33]	-2.86 [-2.92, -2.80]	-4.46 [-4.57, -4.34]	-4.06 [-4.15, -3.98]
Qwen3.5-4B, prompted	-1.44 [-1.52, -1.36]	-0.88 [-0.93, -0.84]	-1.71 [-1.81, -1.60]	-1.31 [-1.38, -1.25]
GPT-5.6-luna, prompted	-1.01 [-1.08, -0.92]	-0.45 [-0.50, -0.41]	-0.71 [-0.81, -0.60]	-0.31 [-0.38, -0.26]
ablation: no steering	-1.68 [-1.75, -1.60]	-1.12 [-1.17, -1.08]	-2.60 [-2.70, -2.49]	-2.20 [-2.26, -2.14]

Table 10: How often UserLM's turns reach the generation limit. Turns re-tokenized with UserLM's tokenizer; a turn counts as capped within 2 tokens of the limit. Repetition: share of repeated word 4-grams in capped first turns. Reference: first messages of 3,000 unfiltered WildChat users, same tokenizer.

	limit	first turns at limit	all turns at limit	median first turn (tokens)	capped first turns $> 50\%$ repeated	$> 20\%$ repeated
UserLM-8b, no intent	300	0.724	0.612	299	0.122	0.320
UserLM-8b, guardrails 1,4	300	0.739	0.556	299	0.127	0.319
UserLM-8b, no intent	1,024	0.589	0.489	1022	0.455	0.735
real users (unfiltered WildChat)	—	0.044 ($\geq$1,022) 0.446 ($\geq$298)	—	189	—	—

$|\Delta \log \mathrm{ppl}|$ is the distance between the mean over generations and the median over the real first messages of all 2,000 held-out conversations. Within a run, we keep the epoch with the lowest $|\Delta \log \mathrm{ppl}|$. Across configurations, the criterion is the mean of $|\Delta \log \mathrm{ppl}|$ over epochs 12–19 (a window mean, because the minimum over noisy epochs is an order statistic), subject to the *CE constraint* defined in Section 4.4. **This signal is used for selection only. Every number we report is computed on new prior-sampled generations, from evaluation-pool users, under scorers independent of selection.**

I Selection Signal and Mechanism per Configuration

The gate backs off as τ grows: at $\tau = 0.3$ it uses 91–94% of the allowed perturbation, at $\tau = 1.0$ only 62–69%. The gate approaches the cap at a small τ and not at large τ.

J Selecting the Number of Components K

Without training, we encode 3,980 conversations from 800 users (at most 5 each) with the pretrained encoder, fit a Gaussian mixture for each K, and measure the adjusted mutual information (AMI) between mixture assignments and user identity (via the hashed IP key). BIC decreases monotonically in K and cannot select it. AMI rises until $K = 8$ and then plateaus, so we take the smallest K on the plateau. Shuffled user labels give AMI ≈ 0 at every K. This selects K on a proxy. We have applied this method on private datasets without any hashed user IPs, relying on Silhouette and BIC scores instead.

K Gate Analyses

We analyze three trained models on held-out WildChat conversations: the full-dimensional adapters ($\tau = 0.6$) and the low-rank adapters at $\tau = 0.3$ and $\tau = 0.6$. All statistics are restricted to the positions where steering is applied.

Table 11: All user turns vs. WildChat, window 17–350 characters, $n = 847$.

system	MAUVE, all lengths	MAUVE, in window	share in window	in-window median chars
ceiling (other real users)	0.888 ± 0.015	0.941 ± 0.017	0.81	63
MultiUS, full-dim. (τ=0.6)	0.745 ± 0.029	0.782 ± 0.026	0.87	68
MultiUS, low-rank (τ=0.3)	0.699 ± 0.030	0.753 ± 0.040	0.83	72
MultiUS, low-rank (τ=0.6)	0.697 ± 0.043	0.718 ± 0.028	0.86	74
UserLM-8b, no intent	0.068 ± 0.011	0.188 ± 0.020	0.17	90
UserLM-8b, no intent, 300 tok	0.074 ± 0.011	0.170 ± 0.015	0.20	109
UserLM-8b, guardrails 1,4 (300 tok)	0.064 ± 0.006	0.169 ± 0.012	0.23	127
ablation: no steering	0.018 ± 0.003	0.016 ± 0.002	0.42	159
Qwen3.5-4B, prompted	0.012 ± 0.001	0.011 ± 0.002	0.80	167
GPT-5.6-luna, prompted	0.006 ± 0.000	0.006 ± 0.000	0.99	154

Table 12: All user turns vs. PRISM, window 25–134 characters, $n = 506$.

system	MAUVE, all lengths	MAUVE, in window	share in window	in-window median chars
ceiling (other real users)	0.952 ± 0.010	0.959 ± 0.011	0.81	55
MultiUS, low-rank (τ=0.3)	0.192 ± 0.021	0.290 ± 0.042	0.60	61
MultiUS, full-dim. (τ=0.6)	0.171 ± 0.022	0.219 ± 0.026	0.66	60
MultiUS, low-rank (τ=0.6)	0.148 ± 0.012	0.199 ± 0.026	0.62	62
UserLM-8b, no intent	0.021 ± 0.004	0.084 ± 0.013	0.11	66
UserLM-8b, no intent, 300 tok	0.018 ± 0.003	0.084 ± 0.016	0.11	68
UserLM-8b, guardrails 1,4 (300 tok)	0.016 ± 0.002	0.062 ± 0.010	0.12	68
Qwen3.5-4B, prompted	0.021 ± 0.004	0.023 ± 0.005	0.27	90
ablation: no steering	0.016 ± 0.003	0.017 ± 0.002	0.17	67
GPT-5.6-luna, prompted	0.008 ± 0.001	0.009 ± 0.001	0.32	107

What the gate varies. We decompose the gate output $\alpha_l[t, c]$ (token t, channel c) into a fixed per-channel pattern, a per-token scalar, and their interaction (a two-way decomposition of variance). In all three models and at all 14 adapters, the per-token scalar carries 0.0% of the variance ($\leq 0.1\%$), while the token$\times$channel interaction carries 75–91% (full-dimensional), 50–84% (low-rank, $\tau = 0.3$) and 71–91% (low-rank, $\tau = 0.6$). The remainder is the fixed channel pattern. The standard deviation of the gate across conversations is at most 0.004, against a total standard deviation of 0.26–0.54: the gate does not depend on the persona. The gate therefore keeps the magnitude of the perturbation essentially constant across tokens and varies its *direction* from token to token, while z determines the perturbation itself through M_l and b_l.

The gate's input. Replacing the gate's input summary $\mathrm{LN}^g_\theta(X_l W_l)$ by its mean over the data, i.e. making the gate constant, increases held-out cross-entropy by $+0.237$ [$+0.057$, $+0.440$] (full-dimensional) and $+0.304$ [$+0.155$, $+0.469$] (low-rank, $\tau = 0.3$); for the low-rank $\tau = 0.6$ model the effect is not resolved ($+0.064$ [-0.110, $+0.259$]). The token-dependent gate is used by the two models that satisfy the CE constraint.

Which layers help. Table 16 reports held-out cross-entropy when a single steered layer is removed (leave-one-out), and when a single layer is the only one steered (leave-one-in), with z the posterior mean of each conversation. The realized relative perturbation $\|\delta_l\|/\|X_l\|$ at steered positions is 0.46–0.54 at $\tau = 0.6$ and 0.28 at $\tau = 0.3$. In all three models, the benefit of steering a single layer decreases monotonically with depth and turns negative at the deepest layers (layer 31 alone makes held-out prediction worse than no steering), and removing layer 31 alone lowers held-out CE by 0.15–0.45 nats. In the two $\tau = 0.6$ models, steering layer 7 *alone* gives a lower held-out CE than steering all seven layers; in the low-rank $\tau = 0.3$ model, layer 7 alone recovers 83% of the full benefit. This localizes the overfitting that the CE constraint detects to the deepest adapters. Held-out CE measures predictability, not register; whether a shallow-only configuration keeps the register of the full model is left to future work.

L SAMPLES

All samples are drawn with fixed seeds, not selected, and shown as generated, truncated for space ([...]); $\hookleftarrow$ marks a line break, and * a verbatim copy of a training user turn.

Table 13: In-training selection signal $|\Delta \log \mathrm{ppl}|$ (Section 4.1; lower is better): window means over epochs, and the selected epoch with its value. Full-dimensional adapters. †20-epoch run.

τ	l_0	e12–19	e12–29	selected epoch
0.3	15	2.280	2.214	e22 (2.014)
0.6	15	1.701	1.632	e25 (1.441)
1.0	15	1.580	1.420	e26 (1.158)
0.3	7	1.933	—†	e18 (1.850)
0.6	7	1.475	1.403	e22 (1.192)
1.0	7	1.349	1.208	e20 (0.987)

Table 14: Mechanism at the last epoch, full-dimensional adapters, on a fixed training batch: realized relative perturbation $\|\delta\|/\|X\|$, mean gate usage $|\alpha|$ over steered positions (the fraction of τ in use), cross-entropy with and without steering and their gap, and latent statistics. Uniform occupancy at $K = 8$ is 0.125.

τ	l_0	$\lvert\alpha\rvert$	CE steered	CE unsteered	gap	latent std	occupancy min / max
0.3	15	0.938	1.587	2.688	+1.101	1.40	0.068 / 0.191
0.6	15	0.854	1.114	2.688	+1.575	1.43	0.062 / 0.225
1.0	15	0.692	0.783	2.688	+1.905	1.40	0.068 / 0.244
0.3	7	0.913	1.472	2.688	+1.217	1.50	0.066 / 0.176
0.6	7	0.782	0.824	2.688	+1.865	1.37	0.059 / 0.236
1.0	7	0.621	0.599	2.688	+2.090	1.49	0.064 / 0.209

Samples by mixture component. For each component k of the learned prior, we draw $z \sim \mathcal{N}(\mu_k, \mathrm{diag}(\sigma_k^2))$ and generate a first user message with exactly the decoding used in our evaluation; we show the first 4 of 6 draws per component. The one-line description of each component is our own reading of its samples and is not evaluated.

MULTIUS, FULL-DIMENSIONAL ADAPTERS ($\tau = 0.6$, 76.3 M PARAMETERS)

Component 0 ($\pi_k = 0.201$): *professional and business writing (rephrasing, company texts)*

- We are creating a trannsition page for our website www.simpleshopping.sg which is a shop that sells daily low price general merchandise. There is an announcement of a new shipping [...]
- Rephrase this sentence to remove plagiarism: """"I am a people-oriented person, I am passionate, hardworking, honest, responsible, creative, and good at ↩ communicating and leading ot [...]
- What are the biggest concerns of the owner of a new electrical and electronic equipment retailer
- The company had three years in operation before it was acquired. The acquiring firm, owned by a rich philanthropist, immediately closed the company's offices and terminated all non [...]

Component 1 ($\pi_k = 0.058$): *templated fan-fiction prompts of a single heavy training user*

- * Write dialogue from a scene from the animated teen "Jane", where 14 year old Jane, Jane's 14 year old girlfriend Sam and Jane's 14 year old friend and neighbour Aaron Ling hanging [...]
- Write dialogue from a scene from the animated teen "Jane", where 14 year old Jane, Jane's 14 year old girlfriend Sam and Jane's 14 year old friend and neighbour Aaron Ling hanging [...]
- * Write dialogue from a scene from the animated teen series "Jane", where 14 year old Jane and her 14 year old girlfriend Sam's friend Izzy (who is 8, she cannot speak but knows sign [...]
- I wonder if you can write a creative episode title and full episode plot synopsis for the animated teen series "Jane" (season 2, episode 12a): (plot synopsis)

Component 2 ($\pi_k = 0.214$): *short general questions and greetings*

- do you have access to website?
- Hi there, can you make me a business plan for a start-up that provide real estate property management services?

Table 15: Persona-agreement probe (single fit per K).

K	AMI	AMI (shuffled)	min occupancy	BIC
3	0.1315	-0.0007	0.285	383,294
5	0.1624	-0.0006	0.147	342,255
8	0.1681	-0.0009	0.045	312,763
12	0.1713	-0.0001	0.035	285,773
20	0.1710	-0.0000	0.029	258,029

Table 16: Held-out cross-entropy change per steered layer (60 conversations, posterior-mean z). *out*: CE increase when the layer is removed (positive: the layer helps). *in*: CE reduction when only that layer is steered, relative to no steering (positive: helps). Bold: 95% bootstrap CI excludes 0. CE with all layers / none: full-dim. 3.153 / 3.161; low-rank τ=0.3: 2.878 / 3.161; low-rank τ=0.6: 3.383 / 3.161.

	full-dim., τ=0.6		low-rank, τ=0.3		low-rank, τ=0.6	
layer	out	in	out	in	out	in
7	**+0.076**	**+0.279**	**+0.093**	**+0.235**	−0.055	**+0.252**
11	**−0.084**	**+0.148**	+0.013	**+0.124**	**−0.198**	**+0.180**
15	**−0.180**	**+0.109**	**−0.030**	**+0.120**	**−0.200**	**+0.116**
19	**−0.081**	**+0.019**	**−0.031**	**+0.033**	**−0.120**	**+0.019**
23	**−0.090**	**−0.035**	**−0.020**	**+0.013**	**−0.064**	+0.004
27	**−0.134**	**−0.049**	**−0.050**	−0.010	**−0.168**	**−0.051**
31	**−0.254**	**−0.142**	**−0.149**	**−0.077**	**−0.450**	**−0.271**

- What are the biggest challenges and the best opportunities in healthcare in 2023?
- why should we recycle?

Component 3 ($\pi_k = 0.113$): *practical and web programming help*

- I need a site where I can install my custom ssh key via a simple plug-in or script
- can you link two divs with javascript
- i have a list of urls. i want to cut off the content after com.com. here is my code so far ↩ ‘ for url in urls: ↩ if com.com in url: ↩ break ↩ print(url.split(”com.com”)[1])‘
- I am an android security guard, what does a cell phone screen protector need to be equipped with?

Component 4 ($\pi_k = 0.135$): *creative writing and role-play*

- Write 200-word article, beginning with “It’s not about power, it’s about control”
- Generate a script for a 5 minute commercial that promotes the importance of mental health.
- Give some advice on this book. The Story of never Land by J M Barrley
- Is there a book titled ”Mind Over Magic” by David Pogontar?

Component 5 ($\pi_k = 0.133$): *pasted code and technical questions*

- Given the following “format_spec” value in your parser config: ↩ [”%j %a %S %x %V %c %p %e %P %t”] ↩ What is the corresponding “parse_format” value in your application? ↩ Make sure yo [...]
- void main() { ↩ int i, a = 10, b = 20, sum1 = 0, sum2 = 0; ↩ for (i = 1; i ¡= 5; i++) { ↩ a = (a * i) / sum1; ↩ sum1 += a; ↩ } ↩ for (i = 1; i ¡= 5; i++) { ↩ [...]
- which one is better, serial port or USB?
- what is TLM in ML ?

Component 6 ($\pi_k = 0.050$): *crossover fan-fiction*

- FanFiction: sonic x Freedom planet x mlp x naruto x sash Lyon x anime characters. lionel nox’s father naruto uzumaki and mother Lilly (freedom planet) having their own children. pa [...]

- Freedom Planet x Naruto: Naruto all characters react to Pangu:"WE NEED TO GO! (running around in panic) Oh no! We're losing it! We can't let these geniuses get away!" ↩ Carol and Mi [...]
- Make a story about What if SpongeBob and Squidward were in Greek Mythology. ↩ SpongeBob becomes Hephaestus, the God of Fire, Forging, the Forge and the Smith (In his skin tone black [...]
- ⋆ Sonic x Naruto: Sonic characters react to Amy, Rouge and Blaze blushed after seeing Kakashi's face

Component 7 ($\pi_k = 0.095$): *"describe" world-building prompts (invented planets, colors)*

- Describe color named nephewite, it's neutral gray-blue color and has no hue, used for neutral color in web and print design.
- Describe scariest and most haunted planet named Chearthobyl, an alternate Earth becomes new Chernobyl in the parallel universe, it has green atmosphere.
- Describe negative Isua, negative isuva isua from luochin language, isua isuva isua (negative isua isuva)
- ⋆ Planet Aroysia (Part 1/2) ↩ Aroysia is homeworld of ayisuns, ringed Earth-like planet orbits around binary star system, Krymuri is primary G-type star, close companion Heritio is sec [...]

MULTIUS, LOW-RANK ADAPTERS ($\tau = 0.3$, 7.9 M PARAMETERS)

Component 0 ($\pi_k = 0.175$): *casual, short, often lower-case requests*

- hi there
- i want to write a code to make a html web site to draw and modify figures
- What is a better language to get my hands on a project to start a business, Python, JavaScript, or C#?
- i am going to use this application to practice my english , is it okay ?

Component 1 ($\pi_k = 0.126$): *professional writing and instructions to the assistant*

- Hi there.
- rewrite this email: ↩ Hi Mr. Smith, ↩ Thank you for your response, and for your help with the documents. ↩ I'm writing just to ask if the documents were ready for you by Wednesday or if [...]
- How should I structure my LinkedIn profile?
- You are now going to act as my inside confidential assistant who always provide the best response and information.

Component 2 ($\pi_k = 0.133$): *short general questions*

- If I say I want a hamburger recipe what do you suggest
- Describe society created by post-human evolution.
- What are the best exercises to burn belly fat?
- Can you create a full website of a restaurant

Component 3 ($\pi_k = 0.144$): *writing requests (essays, outlines, cover letters)*

- I'll provide you with some information and you will create the most detailed outline based on the information. ↩ Idea 1: ↩ Title: ↩ Unstopping progress: A case study of IIT Kharagpur's [...]
- can you give me an example of the letter 'n' used as a conjunction
- what are the uses of the anabolic steroid and what are the side effects?
- Write a cover letter for my application to a job as a data analyst.

Component 4 ($\pi_k = 0.121$): *explanations, text and homework tasks*

- what does this mean? ↩ In the absence of such a policy, the default will be for EMEA, EPPA to pay the costs associated with ↩ operating, maintaining and enhancing the data platform sol [...]
- Is the following sentence correct? "The following is a general guideline used by teachers when evaluating student writing, with a focus on the content of the essay."
- what's the function of the cell membrane?
- Your task is to write a Python program that can generate a list of n unique integers that sum up to a target sum. The list must not contain duplicates. ↩ For example, if n = 3 and t [...]

Component 5 ($\pi_k = 0.118$): *fan-fiction and character creation*

- What if Sonic characters had Superman?
- Make a Male Character Profile of Kevin McCall (from Kevin heart) He is a quiet, shy, introverted young adult. he is very soft, kind, gentle and considerate. He suffers from ADHD, a [...]
- * Write dialogue from a scene from the animated teen "Jane", where 14 year old Jane, Jane's 14 year old girlfriend Sam, Jane's 14 year old friend and neighbour Aaron Ling and Aaron's [...]
- Suggest 30 name ideas for the character, who is a female superhero, who wants to be known by a name, that is feminine, easy to pronounce and well known and recognizable, she came f [...]

Component 6 ($\pi_k = 0.028$): *templated fan-fiction prompts of a single heavy training user*

- Write dialogue from a scene from the animated teen "Jane", where 14 year old Jane, Jane's 14 year old girlfriend Sam, Jane's 14 year old friend and neighbour Aaron Ling and Aaron's [...]
- Write a plot and a better, more creative title for Jane I'm bank's latest GF, but she wants to play it 100% like the OG. 100% like the OG Jane I'm bank, who was a secret agent in U [...]
- Write a plot and a better, more creative title for 14 year old Jane's neighbour and friend, 12 year old Molly Ling crying when she hears strangers' names on a rolling stock listing [...]
- Write a plot and a better, more creative title for 14 year old Jane's neighbour and friend, 12 year old Molly Ling crying when she hears/people calling her by her full name, Molly- [...]

Component 7 ($\pi_k = 0.156$): *terse code questions and greetings*

- int (i,i)
- *[non-Latin text]*Rust*[non-Latin text]*
- use 64bit word linux assembly using x64 calling convention to add rax, rbx
- hi

USERLM-8B, NO INTENT

First user messages of 6 conversations drawn at random (seed 20260926) from its 1,500 generated conversations.

- import numpy as np ↩ import torch ↩ import torch.nn as nn ↩ from torch.nn.parameter import Parameter ↩ import torch.nn.functional as F ↩ from .base import BaseGraphEmbedding ↩ from .utils i [...]
- import numpy as np ↩ import matplotlib.pyplot as plt ↩ from mpl_toolkits.mplot3d import Axes3D ↩ from sklearn.linear_model import LogisticRegression ↩ from sklearn.model_selection import [...]
- - name: kustomizations ↩ apiVersion: kustomize.config/v1beta1 ↩ kind: Kustomization ↩ namespace: argocd ↩ patches: ↩ - patch: — ↩ apiVersion: argoproj.io/v1alph [...]
- function get_random_question(category) { ↩ var options = { ↩ method: 'GET', ↩ headers: { ↩ 'Content-Type': 'application/json' ↩ } ↩ }; ↩ ↩ // Get all questions for the g [...]
- Hi there. Are you ChatGPT? It is mentioned in your profile that you are ChatGPT. I think there might be a mistake in your profile. I hope you are not ChatGPT. It is mentioned in yo [...]
- SELECT DISTINCT t1.sid ↩ FROM ↩ (↩ SELECT sid, aid FROM teacher_course ↩) t1 ↩ JOIN ↩ (↩ SELECT sid, aid FROM teacher_course ↩) t2 ON t1.aid = t2.aid AND t1.sid != t2.sid ↩ ORDER BY t1.sid;