ONE-SHOT CALIBRATION VIA MODEL EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model editing aims to correct errors in large, pretrained models by replacing model generations with user-provided target responses. In practice, however, user feedback often contains multiple plausible answers with varying uncertainty, forming a natural target distribution to not only correct but also calibrate model generation. Existing editors ignore this uncertainty and require a single target answer for each edit. We therefore propose SoftEdit, the first model editor to leverage uncertainty in user responses and perform one-shot calibration during editing. To generalize calibration to unseen samples, SoftEdit uses co-plausibility relationships between answers while preserving real distributional shifts, addressing the new distributional generality–locality tradeoff. Across three language models, three vision-language models, and two large datasets, SoftEdit achieves state-of-the-art performance, ultimately showing that uncertainty in user responses enables generalizable calibration of pretrained models beyond single-answer corrections.[1]

1 INTRODUCTION

Large pretrained text generation models achieve state-of-the-art performance on many tasks, but still make errors and miscalibrated generations after deployment as data distributions, world knowledge, and user needs evolve over time (Ma et al., 2025). Correct and calibrated generations are important for faithful simulations (Hu et al., 2026) and reliable decision support in high-stakes domains (Singhal et al., 2025). To address this gap, many recent methods have been developed to make targeted post hoc model edits. However, they depend on settings where there is one correct answer, such as factual recall. Many real tasks, in contrast, entail multiple plausible answers with varying levels of uncertainty. For example, doctors often provide multiple plausible diagnoses, and different annotators often disagree on subjective tasks. To quickly repair models on inherently uncertain tasks, we need efficient model updates that handle multiple plausible answers and uncertainty between them.

Model editing aims to efficiently make targeted, one-shot updates to large pretrained models using user feedback, while generalizing to unseen samples and preserving unrelated model behaviors (Wang et al., 2024). Recent work has extended from editing factual knowledge in large language models (LLMs) (Mitchell et al., 2021; Meng et al., 2022; Hartvigsen et al., 2023; Gu et al., 2024; Zhang et al., 2025) to editing visually grounded information in vision-language models (VLMs) (Cheng et al., 2023; Huang et al., 2024; Guo et al., 2025; Qiu et al., 2026). However, as shown in Fig. 1, existing model editors assume that each edit is associated with a single target answer, and cannot directly leverage multi-answer feedback and its distributional information to calibrate generation beyond single-answer correction. One way to adapt existing editors would be to collapse distributional feedback into a single target answer. However, this induces miscalibration because maximizing the likelihood of one answer suppresses other plausible answers. Even worse, generalizing this single-answer correction to unseen samples propagates the miscalibration.

We propose soft model editing: given a query, a model, and multiple plausible natural-language answers with uncertainty estimates for each, a model editor produces an updated model that (1) outputs an answer distribution calibrated with the given uncertainty estimates for each plausible answer, (2) generalizes the calibrated distribution to unseen semantically- and distributionally-similar queries, and (3) preserves distributional variations for unseen queries.

Soft model editing poses new challenges. The first challenge is a new soft generality–locality tradeoff: matching the target distribution at the edited query does not ensure calibration on related unseen

[1] All code and data needed to reproduce all findings will be publicly available.

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.

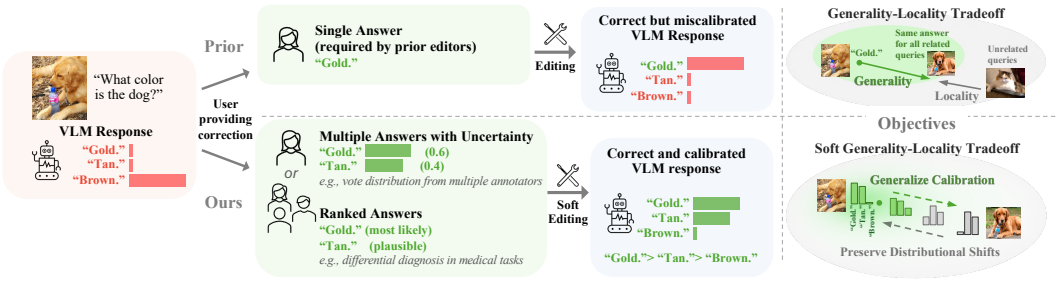

Figure 1: Soft model editing allows uncertainty in user feedback and performs targeted calibration beyond single-answer correction, introducing a new distributional generality–locality tradeoff.

queries, while propagating the same distribution across its semantic neighborhood can overwrite meaningful distributional shifts (Fig. 1). Therefore, there must be a balance between under- and over-generalization. The second challenge is that plausible answers can span multiple tokens. Aligning first-token probabilities (Lee et al., 2023; Ma et al., 2025) calibrates answer selection but does not ensure complete answer generation. Thus, an editor must also preserve accurate generation.

We introduce SoftEdit, the first uncertainty-aware model editor, which calibrates and corrects models using one sample while generalizing the calibration to unseen samples. To enable local calibration from a single query, we adopt a retrieval-based editing framework (Zheng et al., 2023; Guo et al., 2025; Qiu et al., 2026): for each edit, SoftEdit learns and caches a row-sparse weight perturbation alongside a retrieval key. Then if a new query falls close enough to a cached key, the corresponding perturbation is retrieved and temporarily added to the base model weight to generate the answer. We optimize each weight perturbation with three objectives. (1) To generalize calibration beyond an edit, we construct a pairwise co-plausibility graph over the plausible answers and use it to regularize the local weight Jacobian at the edit. This encourages the probabilities of co-plausible answers to change coherently rather than independently as the query moves away from the edit. (2) To prevent overwriting desirable distributional shifts, we regularize the weight perturbation with simple synthetic text-only in-context locality samples constructed from each edit's target answers. (3) To align the answer-level distribution while preserving text generation, we match the model's first-token probabilities to the plausible answers' while optimizing full-text generation of each answer. This ensures stochastic generation of each plausible answer in full text at the target probability, while enabling a fast probability readout through an approximation combining first-token probability with teacher-forced continuation validation without Monte Carlo sampling.

In experiments with three LLMs and three VLMs on two large datasets with multi-answer distributions, SoftEdit achieves state-of-the-art performance in one-shot calibration as well as error correction for model generation. Compared with existing model editors adapted to the same objective, SoftEdit better generalizes calibration to unseen samples, preserves genuine distributional variation, and maintains higher computational efficiency.

In summary, our contributions are as follows:

1. We introduce soft model editing, generalizing post-deployment model editing from single-answer correction to local calibration of large models via multi-answer human feedback.
2. We introduce SoftEdit, a model editing method that performs local calibration while balancing the inherent tradeoff between generalizing and localizing distributional changes.
3. Our extensive experiments across VLMs and LLMs show that SoftEdit achieves state-of-the-art performances with high computational efficiency.

2 Related Work

Post-training Model Editing. Model editing enables efficient, targeted corrections to large pre-trained models using individual human feedback, aiming to generalize these updates to unseen samples (generality) while preserving unrelated behaviors (locality). Balancing generality and locality becomes harder as edits accumulate and interfere sequentially. Early editors rely on fine-tuning,

which can cause model collapse (Sinitsin et al., 2020), while later methods such as MEND (Mitchell et al., 2021), ROME (Meng et al., 2022), and AlphaEdit (Fang et al., 2025) use more targeted weight updates to better preserve model behaviors. Retrieval-based editors such as GRACE (Hartvigsen et al., 2023) and IKE (Zheng et al., 2023) instead perform edits without modifying pretrained model weights, often achieving strong locality but conservative generalization. Recent work has extended model editing to VLMs (Cheng et al., 2023; Huang et al., 2024; Zhang et al., 2025), aiming to edit visually grounded information while balancing multimodal generality and locality (Zeng et al., 2025; Guo et al., 2025; Qiu et al., 2026). However, existing model editors require users to provide a single ground-truth answer for each model error. They neither account for uncertainty over multiple plausible answers in realistic human feedback nor leverage this distributional information to calibrate model generation beyond single-answer correction. Collapsing to a single answer can further propagate miscalibration to related inputs. Our work instead allows for uncertainty over multiple plausible answers and targets distributional calibration beyond single-answer correction.

Post-training Model Calibration. Large pretrained models can be miscalibrated at inference, often overconfidently favoring a single response and failing to capture realistic uncertainty over multiple plausible answers (Lee et al., 2023; Cheng et al., 2024; Ma et al., 2025; Inoshita et al., 2026; Jiang & Liang, 2026; Ni et al., 2026; Wu et al., 2026). Calibrating model uncertainty helps capture real-world ambiguity (Peterson et al., 2019; Hüllermeier & Waegeman, 2021; Fleisig et al., 2024) and supports high-stakes domains such as medicine, where uncertainty over diagnostic predictions provides crucial information for decision-support (Mimori et al., 2021; Jung, 2025; Singhal et al., 2025). Traditional calibration methods apply global post-hoc transformations using a held-out dataset of single-labeled samples drawn from the target distribution to improve aggregate calibration (Platt et al., 1999; Zadrozny & Elkan, 2001; Guo et al., 2017). However, global calibration can hide local miscalibration and does not ensure the predictive distribution matches the target distribution for individual samples (Luo et al., 2022; Joy et al., 2023). More flexible approaches include Dirichlet calibration (Kull et al., 2019) and adaptive temperature scaling (Joy et al., 2023), but they also require a curated held-out calibration dataset and cannot incorporate real-time human feedback to update a pretrained model. Our work targets this gap by introducing one-shot calibration, which efficiently calibrates individual samples and generalizes the calibration to unseen samples.

3 One-Shot Calibration via Model Editing

3.1 Problem Definition

Let $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$ denote an edit, where x is an input query to a pretrained model and $\mathcal{Y}_K = (y_1, \ldots, y_K)$ is the corresponding set of K plausible human-provided answers. The K answers follow an answer-level target distribution $\mathbf{p}_K(\cdot \mid x) = (p_1, \ldots, p_K)$ estimated from human feedback, where p_k denotes the probability that a human provides y_k as the answer. For a language model, x is a text question. For a vision-language model, x pairs an image with a text question.

We aim to develop a model editor g that takes in an edit e and a pretrained model f, and outputs a locally calibrated model $f_{\text{new}} = g(e, f)$. Let $\widehat{\mathbf{p}}_K(\cdot \mid x; f_{\text{new}})$ denote the model's distribution over the K human-provided answers under stochastic generation. After editing, f_{new} should 1) align $\widehat{\mathbf{p}}_K$ with the human target distribution $\mathbf{p}_K$, such that each answer y_k is generated in full text with target probability p_k, 2) generalize the calibrated distribution to unseen queries that are semantically and distributionally similar to the edit, and 3) preserve real distributional shifts. Note that conventional model editing is a special case of our one-shot calibration problem: when the target distribution assigns probability one to a single ground-truth answer, one-shot local calibration reduces to standard single-answer correction.

3.2 SoftEdit

We propose SoftEdit, the first model editor targeting one-shot model calibration beyond single-answer correction. To enable localized editing, SoftEdit adopts a retrieval-based framework (Zheng et al., 2023; Guo et al., 2025; Qiu et al., 2026) and stores a local weight perturbation matrix ΔW_e for each edit. Rather than storing a fixed answer or target distribution, ΔW_e captures how the target distribution should vary across the edit's semantic neighborhood. At inference, when a new query

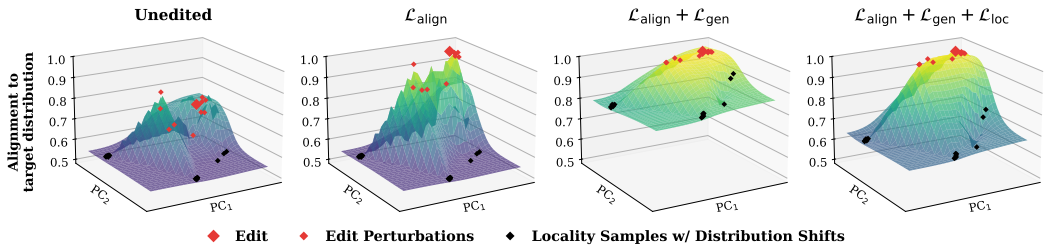

Figure 2: Visualization of changes in the local distribution field of a real edit from VQAv2 as SoftEdit objectives are progressively applied to edit LLaVA-1.5-7B. The xy-plane is a PCA projection of neighbors' last-token hidden states; the z-axis shows the distributional alignment towards the edit's target distribution. SoftEdit ($\mathcal{L}_{\text{align}} + \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{loc}}$) calibrates the edit (alignment $\rightarrow 1$ at the large red point), generalizes calibration to distributionally equivalent neighbors (alignment $\rightarrow 1$ at other red points) without overwriting distributional shifts in locality samples (low alignment to edit target at black points).

falls within the semantic neighborhood of an edit, SoftEdit retrieves ΔW_e and temporarily applies to original model weight to generate an updated answer using a calibrated distribution for this query.

3.2.1 Learning ΔW_e: Soft Distributional Generality–Locality Tradeoff

For each edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, we learn a local weight perturbation ΔW_e with the objective:

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \lambda_{\text{gen}}\, \mathcal{L}_{\text{gen}} + \lambda_{\text{loc}}\, \mathcal{L}_{\text{loc}},$$

where $\mathcal{L}_{\text{align}}$ aligns the model's answer-level distribution with the target distribution $\mathbf{p}_K$ for the edit e. $\mathcal{L}_{\text{gen}}$ encourages this calibration to generalize to semantically similar unseen queries that share the target distribution $\mathbf{p}_K$, while $\mathcal{L}_{\text{loc}}$ preserves real distributional shifts. λ_{gen} and λ_{loc} balance the loss terms using standard loss-scale normalization (Kendall et al., 2018).

We only edit the final vocabulary projection layer, where $W \in \mathbb{R}^{|\mathcal{V}| \times d}$ maps the last hidden state $h \in \mathbb{R}^d$ to logits $z \in \mathbb{R}^{|\mathcal{V}|}$. For edit e, let $\mathcal{S}_e$ be the set of unique token indices that appear in the K target answers, including the end-of-sequence token. SoftEdit learns a row-sparse perturbation $\Delta W_e \in \mathbb{R}^{|\mathcal{S}_e| \times d}$ only for these rows by optimizing $\mathcal{L}$, and stores it with their token indices and the retrieval key. Let $\oplus_{\mathcal{S}_e}$ denote $\mathcal{S}_e$-indexed addition. The updated layer weight is $W' = W \oplus_{\mathcal{S}_e} \Delta W_e$. Editing the final projection weight keeps ΔW_e compact and fast to optimize: only W's rows indexed by $\mathcal{S}_e$ are modified, and $|\mathcal{S}_e| \ll |\mathcal{V}|$. The hidden states required by $\mathcal{L}$ can therefore be computed once and reused across all optimization steps, avoiding repeated forward passes.

$\mathcal{L}_{\text{align}}$: Local targeted calibration. To align the model distribution $\widehat{\mathbf{p}}_K$ with the human-provided distribution $\mathbf{p}_K$, while preserving full-text generation, we supervise both answer selection and answer completion. Each answer y_k consists of a token sequence $(y_{k,1}, \ldots, y_{k,L_k})$ of length L_k, where $y_{k,1}$ is its first token and y_{k,L_k} is the appended end-of-sentence token. We use its first-token probability to represent how likely the model is to select that answer, and optimize a continuation term—the length-normalized negative log-likelihood of the remaining tokens—to ensure that the complete answer is generated correctly. Let h_0 denote the edit query x's final hidden state before the final projection layer. The alignment loss is given by:

$$\mathcal{L}_{\text{align}} = \underbrace{D_{\text{KL}}\big(\mathbf{p}_K \,\big\|\, \widehat{\mathbf{p}}_K(h_0)\big)}_{\text{first-token alignment}} + \underbrace{\frac{1}{K}\sum_{k=1}^{K}\left[-\frac{1}{L_k - 1}\sum_{t=2}^{L_k} \log p_\theta(y_{k,t} \mid x,\, y_{k,<t})\right]}_{\text{full-answer continuation}},$$

where $p_\theta(y_{k,t} \mid x, y_{k,<t})$ is the model's probability of generating the t-th token conditioned on all preceding tokens. As shown by a real edit example in Fig. 2, $\mathcal{L}_{\text{align}}$ calibrates the model to the target distribution at the edit. However, $\mathcal{L}_{\text{align}}$ alone does not ensure that this calibration generalizes to semantically and distributionally equivalent samples. High-entropy target distributions can also place edits near decision boundaries, where small perturbations may induce unstable changes in the predicted distribution (He et al., 2018; Pearce et al., 2021).

$\mathcal{L}_{\text{gen}}$: Generalizing calibration. To generalize calibration beyond the single edit query x, we leverage the co-plausibility relationships between answers. Specifically, we construct a pairwise

co-plausibility graph over the K answers and apply graph-Laplacian regularization (Belkin et al., 2006) to the Jacobian of their edited first-token probabilities with respect to the hidden state h, evaluated at h_0. The edge weight between answers y_i and y_j is $A_{ij} = p_i\, p_j$ ($i \neq j$, $A_{ii} = 0$), so co-plausible answers with higher probabilities are more strongly connected. Letting $J_i = \partial \widehat{p}_i / \partial h \big|_{h_0}$ be the gradient of the edited readout for answer y_i, the generalization loss is

$$\mathcal{L}_{\text{gen}} = \sum_{i,j} A_{ij} \, \|J_i - J_j\|_2^2.$$

This graph-Laplacian regularization encourages co-plausible answers' probabilities to change coherently as the input moves away from the edit, helping preserve the target answers' relative ordering and distribution in a neighborhood of h_0. As shown in Fig. 2, adding $\mathcal{L}_{\text{gen}}$ generalizes the calibrated distribution to samples that are semantically and distributionally equivalent to the edit. However, imposing the same distribution across the entire semantic neighborhood can erase genuine distributional shifts in semantically similar but distributionally different samples.

$\mathcal{L}_{\text{loc}}$: Preserving distributional shifts. Within the semantic neighborhood of an edit e, unseen queries may share high semantic similarity due to closely related context or content, but semantic proximity does not imply distributional similarity: small differences in answer-relevant details can substantially shift the relative probabilities of the target answers. To prevent SoftEdit from overwriting distributional shifts, we regularize the learning of ΔW_e using synthetic text-only locality samples constructed from each edit's target answers through a very simple procedure.

Specifically, for each answer y_k, we synthesize an unambiguous declarative prompt x_k^{loc} by prepending "The answer to [question x] is y_k." to the original question x, and pairing it with a one-hot target distribution where we assign probability 1 to y_k. Each locality sample $(x_k^{\text{loc}}, y_k; (p_k = 1))$ therefore represents a semantically similar query where one target answer should clearly dominate. Optimizing distributional alignment on these locality samples encourages the edited model to preserve genuine distributional shifts. We use the same first-token alignment and continuation structure as $\mathcal{L}_{\text{align}}$, applied to each locality sample:

$$\mathcal{L}_{\text{loc}} = \frac{1}{K} \sum_{k=1}^{K} \left[\underbrace{D_{\text{KL}}\big(\mathbf{p}_k^{\text{loc}} \,\big\|\, \widehat{\mathbf{p}}_K(h_k^{\text{loc}})\big)}_{\text{first-token alignment}} + \underbrace{\frac{1}{K} \sum_{j=1}^{K} \left[-\frac{1}{L_j - 1} \sum_{t=2}^{L_j} \log p_\theta\big(y_{j,t} \mid x_k^{\text{loc}},\, y_{j,<t}\big) \right]}_{\text{full-answer continuation}} \right],$$

where $\mathbf{p}_k^{\text{loc}}$ is the one-hot target on y_k and h_k^{loc} is the hidden state at x_k^{loc}. Because the anchors are purely textual, this approach applies directly to not only LLMs but also VLMs without requiring answer-specific counterfactual images. As shown in Fig. 2, by incorporating both $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{loc}}$, SoftEdit generalizes the calibrated distribution while preserving genuine distributional shifts. More examples are provided in Appendix Fig. 7.

Balancing weights. λ_{loc} and λ_{gen} are set automatically via loss-scale normalization (Kendall et al., 2018), computed once on the first 100 accumulated edits using the unedited model. Specifically, $\lambda_{\text{loc}} = \bar{\mathcal{L}}_{\text{gen}} / \bar{\mathcal{L}}_{\text{loc}}$ equalizes the scales of the two regularization terms, while λ_{gen} scales the sum of them relative to $\mathcal{L}_{\text{align}}$ to prevent negligible regularization. Details are provided in Appendix B.1.

3.2.2 Storing and Retrieving ΔW_e

Weight-update editors (Mitchell et al., 2021; Meng et al., 2022; Fang et al., 2025) capture nuanced output distribution changes as inputs change, but shared parameter updates cause interference and catastrophic forgetting as edits accumulate. Retrieval-based editors (Zheng et al., 2023; Hartvigsen et al., 2023; Qiu et al., 2026) improve local preservation by restricting each edit to its semantic neighborhood, but store a fixed answer or target distribution that cannot adapt to distributional shifts within that neighborhood. SoftEdit leverages both strategies by storing the learned ΔW_e for each edit and retrieving it to provide an updated distributional mapping for its semantically similar queries.

Storing ΔW_e. After learning ΔW_e, we store it in a codebook together with a retrieval key constructed from the embedding representation $\mathcal{E}(x)$ of the edit query x. At inference, this key is used to retrieve the relevant ΔW_e when a new query's embedding falls within the edit's semantic neighborhood. We follow prior retrieval-based editors for constructing the retrieval key. For VLMs,

we adopt the dual embedding for multimodal queries from ReasonEdit (Qiu et al., 2026), combining the vision encoder's embedding of the image and a sentence encoder's embedding of the question using topology-balanced multimodal weighting. For LLMs, following IKE and BalancEdit (Zheng et al., 2023; Guo et al., 2025), we use the last-token hidden-state embedding from the last layer.

Gated retrieval of ΔW_e at inference. At inference, a new query is encoded into the same embedding space and matched to the nearest codebook key by L_2 distance. Following prior retrieval-based editors (Qiu et al., 2026), we use the distances among codebook keys to estimate how far semantically unrelated queries typically lie in the embedding space. Retrieval is rejected when the distance from a new query to its nearest key exceeds a percentile γ of this distance distribution. We automatically determine this rejection percentile using a small batch of accumulated edits: we generate semantics-preserving augmentations of these edits and sweep candidate percentiles of the key-distance distribution to choose the lowest percentile that preserves accurate retrieval of each edit for its augmentations. Further details are provided in Appendix B.2.

Algorithm 1 SoftEdit

=== Editing ===

Input: $f(\cdot; W)$, pretrained model
Input: $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, edit
Input: $\mathcal{C}$, codebook
Step 1: Learn ΔW_e
Learn row-sparse ΔW_e by minimizing
$\mathcal{L}_{\text{align}} + \lambda_{\text{gen}}\mathcal{L}_{\text{gen}} + \lambda_{\text{loc}}\mathcal{L}_{\text{loc}}$
Step 2: Store edit
$\mathrm{K}_e \leftarrow \mathcal{E}(x)$
$\mathcal{C} \leftarrow \mathcal{C} \cup \{(\mathrm{K}_e, \Delta W_e)\}$
Output: updated codebook $\mathcal{C}$

=== Inference ===

Input: x', new query
Step 1: Retrieve edit
$\mathrm{K}' \leftarrow \mathcal{E}(x')$
Calculate $\|\mathrm{K}' - \mathrm{K}_e\|_2$
Retrieve nearest $(\mathrm{K}_e, \Delta W_e) \in \mathcal{C}$
Step 2: Apply or reject edit
if $\|\mathrm{K}' - \mathrm{K}_e\|_2 \le \gamma$ **then**
$f_{\text{new}}(\cdot) \leftarrow f(\cdot; W + \Delta W_e)$
else
$f_{\text{new}}(\cdot) \leftarrow f(\cdot; W)$
end if
Output: $y' = f_{\text{new}}(x')$

4 Experiments

4.1 Evaluation Metrics

To evaluate distributional calibration for an input query x, we compare the edited model's answer-level distribution $\widehat{\mathbf{p}}_K(\cdot \mid x;\ f_{\text{new}})$ with the target distribution $\mathbf{p}_K$. A standard Monte Carlo estimate samples many full generations and computes answer frequencies (Sorensen et al., 2022; Lee et al., 2023), but is prohibitively expensive at scale. We therefore introduce a fast answer-level probability estimator using the model's first-token probability $p_\theta(t_{k,1} \mid x)$ combined with a binary continuation gate $c_k(x) \in \{0, 1\}$, which verifies whether the model correctly completes the full answer via teacher forcing: $\widehat{p}(y_k \mid x) = p_\theta(t_{k,1} \mid x) \cdot c_k(x)$. Full details are provided in Appendix A.1. We then measure a gated Jensen–Shannon distance: $\mathrm{JS}_{\text{gate}} = \sqrt{\mathrm{JSD}(\widehat{\mathbf{p}}_K \| \mathbf{p}_K)} \in [0, 1]$. Lower values indicate closer distributional alignment. Based on the problem definition, we evaluate an editor's one-shot calibration performance on three properties: *alignment, soft generality, and soft locality*, as well as standard model editing performance: *reliability, generality, and locality*. Unseen samples are used to evaluate generality and locality metrics, and construction details are provided in Appendix A.2.

Alignment (Align). The edited model should match the human-provided target distribution $\mathbf{p}_K$ on each edit input. $\text{Align} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\text{gate}}(\mathbf{p}_K,\ \widehat{\mathbf{p}}_K(\cdot \mid x;\ f_{\text{new}}))\right]$.

Soft Generality (SGen). The edited model should generalize the calibrated distribution to unseen semantically and distributionally similar queries. Let $R(x)$ denote variants of x that preserve the target distribution $\mathbf{p}_K$. Then, $\text{SGen} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\text{gate}}(\mathbf{p}_K,\ \widehat{\mathbf{p}}_K(\cdot \mid R(x);\ f_{\text{new}}))\right]$. We evaluate text variants (SGen-T) using question rephrasing and, for VLMs, image variants (SGen-I) using generated semantically and distributionally similar images following prior studies (Appendix A.2).

Soft Locality (SLoc). The edited model should preserve genuine distribution shifts in unseen samples that are semantically similar to an edit query x, rather than overwriting them with $\mathbf{p}_K$. Let $R_{\text{loc}}(x)$ denote related inputs whose target distribution $\mathbf{p}_K^{\text{loc}} \neq \mathbf{p}_K$. $\text{SLoc} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\text{gate}}\big(\mathbf{p}_K^{\text{loc}},\ \widehat{\mathbf{p}}_K(\cdot \mid R_{\text{loc}}(x);\ f_{\text{new}})\big)\right]$.

Model Editing Metrics. Given model editing is a special case of our setup, we also measure traditional model editing metrics: Reliability measures whether the edited model generates a correct answer from $\mathcal{Y}_K$ on the edit input: $\text{Acc} = \mathbb{E}_{e \in D_{\text{edit}}} \mathbf{1}\{f_{\text{new}}(x) \in \mathcal{Y}_K\}$. Generality measures correctness on unseen semantically equivalent inputs: $\mathbb{E}_{e \in D_{\text{edit}}} \mathbf{1}\{f_{\text{new}}(R(x)) \in \mathcal{Y}_K\}$. Locality measures whether the original prediction is preserved on unrelated samples D_u: $\mathbb{E}_{x \in D_u} \mathbf{1}\{f_{\text{new}}(x) = f(x)\}$.

Editor	LLaVA-1.5-7B (4,436 errors)								InstructBLIP-7B (4,625 errors)								Qwen3-4B (4,157 errors)							
	Align	SGen-I	SGen-T	SLoc	Acc	Gen-I	Gen-T	Loc	Align	SGen-I	SGen-T	SLoc	Acc	Gen-I	Gen-T	Loc	Align	SGen-I	SGen-T	SLoc	Acc	Gen-I	Gen-T	Loc
Unedited	0.08	0.16	0.08	0.23	0.02	0.38	0.12	1.00	0.17	0.23	0.17	0.29	0.02	0.35	0.12	1.00	0.11	0.22	0.12	0.37	0.01	0.39	0.11	1.00
FT	0.52	0.47	0.48	0.41	0.96	0.86	0.87	0.30	0.53	0.48	0.50	0.45	0.95	0.87	0.90	0.47	0.53	0.45	0.49	0.41	0.96	0.82	0.90	0.43
MEND	0.12	0.15	0.12	0.15	0.25	0.32	0.23	0.40	0.11	0.14	0.11	0.16	0.17	0.24	0.17	0.29	0.08	0.09	0.08	0.09	0.16	0.20	0.16	0.21
GRACE	0.53	0.36	0.36	0.35	0.91	0.72	0.62	**0.97**	0.53	0.37	0.40	0.40	0.87	0.64	0.61	**0.98**	0.54	0.26	0.31	0.40	0.90	0.47	0.50	**0.99**
BalancEdit	0.23	0.27	0.21	0.31	0.39	0.49	0.36	0.51	0.23	0.26	0.21	0.31	0.36	0.44	0.34	0.59	0.19	0.24	0.18	0.32	0.32	0.44	0.31	0.56
ReasonEdit	0.40	0.40	0.37	0.36	0.97	0.96	0.93	0.54	0.40	0.42	0.39	0.38	0.87	0.88	0.84	0.71	0.53	0.53	0.51	0.50	**0.99**	0.98	0.95	0.64
FT-Soft	0.74	0.59	0.61	0.48	0.98	0.89	0.91	0.30	0.74	0.58	0.64	0.52	0.98	0.90	0.94	0.48	0.79	0.55	0.66	0.46	**0.99**	0.85	0.94	0.26
MEND-Soft	0.13	0.15	0.12	0.16	0.23	0.33	0.23	0.48	0.13	0.16	0.13	0.17	0.26	0.33	0.26	0.44	0.06	0.07	0.06	0.07	0.13	0.16	0.13	0.20
GRACE-Soft	0.91	0.50	0.56	0.41	0.90	0.72	0.62	**0.97**	0.87	0.48	0.59	0.47	0.87	0.64	0.61	**0.98**	**0.92**	0.29	0.45	0.42	0.90	0.47	0.50	**0.99**
BalancEdit-Soft	0.22	0.26	0.20	0.30	0.36	0.45	0.33	0.44	0.23	0.26	0.21	0.30	0.33	0.43	0.33	0.46	0.21	0.26	0.20	0.34	0.32	0.46	0.31	0.66
ReasonEdit-Soft	0.51	0.50	0.49	0.43	**0.99**	**0.98**	0.96	0.53	0.51	0.53	0.48	0.47	0.91	0.92	0.86	0.79	0.56	0.55	0.55	0.60	**0.99**	**0.99**	**0.97**	0.66
SoftEdit	**0.97**	**0.78**	**0.92**	**0.61**	**0.99**	0.92	**0.98**	0.92	**0.98**	**0.82**	**0.88**	**0.68**	**1.00**	**0.98**	**0.99**	0.78	0.89	**0.67**	**0.82**	**0.62**	0.94	0.88	0.93	0.95

(1) VLM Editing on VQAv2 (19,186 samples)

Editor	Llama-3.1-8B (8,326 errors)						Mistral-7B (7,782 errors)						Qwen2.5-7B (7,207 errors)					
	Align	SGen	SLoc	Acc	Gen	Loc	Align	SGen	SLoc	Acc	Gen	Loc	Align	SGen	SLoc	Acc	Gen	Loc
Unedited	0.04	0.04	0.07	0.01	0.03	1.00	0.03	0.03	0.11	0.02	0.07	1.00	0.04	0.06	0.17	0.00	0.10	1.00
FT	0.48	0.41	0.29	0.98	0.86	0.25	0.48	0.37	0.25	0.98	0.80	0.34	0.48	0.40	0.25	0.97	0.83	0.20
MEND	0.14	0.11	0.07	0.40	0.32	0.10	0.05	0.03	0.03	0.19	0.13	0.08	0.04	0.03	0.02	0.12	0.10	0.05
GRACE	0.49	0.27	0.15	0.34	0.21	0.96	0.50	0.04	0.11	0.59	0.09	**1.00**	0.48	0.26	0.20	0.58	0.35	**0.98**
IKE	0.08	0.07	0.09	0.23	0.18	0.59	0.08	0.06	0.08	0.26	0.20	0.49	0.36	0.30	0.23	0.85	0.71	0.50
AlphaEdit	0.32	0.20	0.19	0.69	0.44	0.25	0.42	0.23	0.24	0.88	0.46	0.38	0.28	0.10	0.21	0.58	0.21	0.32
FT-Soft	0.81	0.55	0.37	**1.00**	0.87	0.23	0.75	0.46	0.31	**1.00**	0.83	0.33	0.80	0.52	0.31	**0.99**	0.85	0.27
MEND-Soft	0.19	0.15	0.13	0.51	0.43	0.15	0.04	0.03	0.06	0.14	0.11	0.21	0.06	0.05	0.06	0.19	0.15	0.14
GRACE-Soft	**0.95**	0.44	0.17	0.35	0.22	**0.97**	0.92	0.05	0.12	0.57	0.09	**1.00**	0.87	0.36	0.20	0.58	0.35	**0.98**
IKE-Soft	0.07	0.07	0.10	0.20	0.17	0.58	0.08	0.08	0.07	0.25	0.23	0.49	0.34	0.30	0.35	0.80	0.71	0.50
AlphaEdit-Soft	0.38	0.22	0.21	0.67	0.45	0.27	0.40	0.21	0.25	0.68	0.42	0.31	0.32	0.10	0.23	0.62	0.20	0.36
SoftEdit	**0.95**	**0.72**	**0.47**	0.98	**0.94**	0.76	**0.93**	**0.70**	**0.52**	0.98	**0.95**	0.69	**0.93**	**0.73**	**0.52**	0.98	**0.94**	0.77

(2) LLM Editing on ProtoQA (9,762 samples)

Table 1: Batch editing results on VLMs and LLMs using the VQAv2 and ProtoQA datasets.

4.2 Experimental Setup

Datasets. We evaluate editors on two multi-answer question–answering benchmarks with multi-annotator vote distributions. (1) VQAv2 (Goyal et al., 2017) contains visual questions answered by 10 annotators. We use the vote distribution over free-form answers for each sample and retain 19,186 samples for VLM editing. (2) ProtoQA (Boratko et al., 2020) contains commonsense questions with free-form answers clustered into canonical groups. We use vote distribution for each edit and retain 9,762 samples for LLM editing. Using a pretrained model's predictions on each dataset, we construct the edit set (error set) D_{edit} from incorrectly answered samples $(f(x) \notin \mathcal{Y}_K)$, and unrelated set D_u from correctly-answered ones $(f(x) \in \mathcal{Y}_K)$. Data processing details are provided in Appendix C.1.

Edited Models. We edit three popular VLMs: LLaVA-1.5-7B (Liu et al., 2024), InstructBLIP-7B (Dai et al., 2023), and Qwen3-VL-4B-Instruct (Yang et al., 2025) and three popular LLMs: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), and Qwen2.5-7B-Instruct (Qwen et al., 2025). Details on these models are provided in Appendix C.2.

Baseline Editors. We compare SoftEdit with five VLM editors and five LLM editors adapted for one-shot calibration (denoted [editor]-soft). For weight-updating editors: FT (Fine-tuning), MEND (Mitchell et al., 2021), AlphaEdit (Fang et al., 2025), GRACE (Hartvigsen et al., 2023), and BalancEdit (Guo et al., 2025), we replace their single-answer cross-entropy loss with $\mathcal{L}_{\text{align}}$. For retrieval-based methods, ReasonEdit (Qiu et al., 2026) and IKE (Zheng et al., 2023), we store and retrieve the ranked list of plausible answers. Details on these editors are provided in Appendix C.3.

4.3 Main Results

Comparison to prior editors. We first compare SoftEdit with state-of-the-art editors adapted to model calibration in static editing, where all edits are available for each editor. Soft editors use the distribution over all plausible answers as the target, while their standard counterparts use a one-hot target on the majority-vote answer. As shown in Table 1, all soft editors outperform their non-soft counterparts in calibration (Align, SGen, and SLoc) while maintaining comparable editing performance, suggesting that calibration constitutes a broader objective beyond single-answer correction. SoftEdit consistently outperforms other soft editors in Align, SGen, and SLoc, showing strongest calibration generalization while preserving genuine distribution shifts. Besides calibration, SoftEdit shows comparable model editing performance to state-of-the-art editors, with high Acc, Gen, and

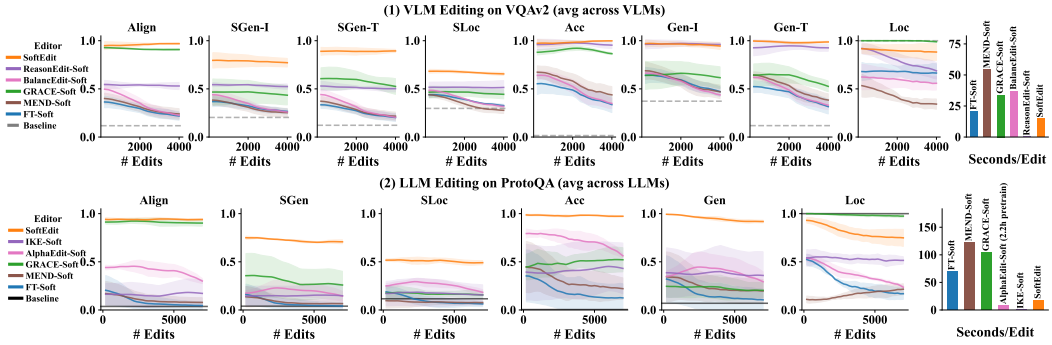

Figure 3: Sequential one-shot calibration performance and computational efficiency across soft VLM and LLM editors. Results are averaged across edited pretrained models.

Editor	LLaVA-1.5-7B (4,436 errors)								InstructBLIP-7B (4,625 errors)								Qwen3-4B (4,157 errors)							
	Rank-Align	Rank-SGen-I	Rank-SGen-T	Rank-SLoc	Acc	Gen-I	Gen-T	Loc	Rank-Align	Rank-SGen-I	Rank-SGen-T	Rank-SLoc	Acc	Gen-I	Gen-T	Loc	Rank-Align	Rank-SGen-I	Rank-SGen-T	Rank-SLoc	Acc	Gen-I	Gen-T	Loc
Unedited	0.15	0.26	0.17	0.33	0.02	0.38	0.12	1.00	0.15	0.27	0.18	0.36	0.02	0.35	0.12	1.00	0.15	0.29	0.18	0.43	0.01	0.39	0.11	1.00
FT-Soft	0.78	0.60	0.63	0.50	0.98	0.88	0.91	0.34	0.79	0.61	0.67	0.54	0.98	0.90	0.93	0.50	0.80	0.54	0.66	0.47	**0.99**	0.85	**0.94**	0.25
MEND-Soft	0.19	0.22	0.18	0.26	0.23	0.30	0.22	0.40	0.13	0.14	0.13	0.15	0.12	0.13	0.12	0.10	0.14	0.15	0.13	0.17	0.15	0.17	0.14	0.18
GRACE-Soft	**0.95**	0.55	0.60	0.47	0.88	0.74	0.62	**0.98**	0.83	0.48	0.56	0.50	0.86	0.64	0.61	**0.99**	**0.96**	0.34	0.45	0.46	0.89	0.47	0.50	**0.99**
BalancEdit-Soft	0.26	0.31	0.24	0.36	0.36	0.45	0.33	0.44	0.26	0.29	0.25	0.36	0.34	0.43	0.33	0.52	0.24	0.30	0.23	0.39	0.30	0.45	0.30	0.61
SoftEdit	0.94	**0.72**	**0.84**	**0.59**	**0.99**	**0.92**	**0.98**	0.92	**0.96**	**0.70**	**0.78**	**0.63**	**0.99**	**0.97**	**0.99**	0.82	0.92	**0.65**	**0.78**	**0.63**	0.95	**0.89**	0.93	0.96

(1) VLM Editing on VQAv2 (19,186 samples)

Editor	Llama-3.1-8B (8,326 errors)						Mistral-7B (7,782 errors)						Qwen2.5-7B (7,207 errors)					
	Rank-Align	Rank-SGen	Rank-SLoc	Acc	Gen	Loc	Rank-Align	Rank-SGen	Rank-SLoc	Acc	Gen	Loc	Rank-Align	Rank-SGen	Rank-SLoc	Acc	Gen	Loc
Unedited	0.14	0.14	0.17	0.01	0.03	1.00	0.12	0.12	0.18	0.02	0.07	1.00	0.14	0.16	0.26	0.00	0.10	1.00
FT-Soft	0.95	0.67	0.54	**1.00**	0.89	0.21	0.91	0.65	0.51	**1.00**	0.85	0.26	0.93	0.69	0.49	**1.00**	0.87	0.24
MEND-Soft	0.09	0.09	0.07	0.06	0.05	0.03	0.07	0.07	0.00	0.00	0.00	0.00	0.07	0.07	0.02	0.02	0.02	0.00
GRACE-Soft	**0.97**	0.56	0.32	0.35	0.24	**0.98**	**0.96**	0.15	0.16	0.57	0.09	**1.00**	**0.97**	0.48	0.33	0.57	0.37	**0.99**
AlphaEdit-Soft	0.48	0.32	0.33	0.64	0.45	0.27	0.53	0.33	0.45	0.68	0.45	0.32	0.44	0.20	0.34	0.61	0.21	0.37
SoftEdit	**0.97**	**0.70**	**0.59**	0.98	**0.94**	0.75	**0.96**	**0.67**	**0.70**	0.98	**0.95**	0.69	0.94	**0.71**	**0.67**	0.98	**0.95**	0.79

(2) LLM Editing on ProtoQA (9,762 samples)

Table 2: Batch editing VLMs and LLMs when human feedback provides only ranked answers.

Loc indicating accurate answer generation on edits and unseen samples. GRACE-Soft achieves the second-best alignment by moving each edit activation toward the target distribution, but optimizing $\mathcal{L}_{\text{align}}$ alone does not generalize calibration to unseen samples, yielding low SGen. In contrast, SoftEdit's graph-Laplacian generality term improves SGen. Batch fine-tuning (FT) achieves the second-best SGen–SLoc tradeoff by learning global calibration across thousands of edits, but its lower edit alignment indicates weaker reliability and precision for local targeted calibration.

Sequential one-shot calibration and computing efficiency. We next examine sequential editing, where each editor sequentially calibrates one edit at a time to repeatedly update a pretrained model. We evaluate the model every 200 edits on 50 random previous edits. As shown in Fig. 3, SoftEdit maintains consistently better and more stable one-shot calibration performance than other soft editors over thousands of edits. This indicates that $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{loc}}$ continuously balance the distributional generality–locality tradeoff as edits arrive in real time. As shown in right panels, SoftEdit also achieves high computational efficiency by editing only the final projection layer. As expected, weight-updating editors exhibit catastrophic forgetting and model degradation after a few hundred edits while FT performs notably worse one-shot than batched.

Calibration with ranked answers. Finally, we study one-shot calibration when given only a ranking of plausible answers without explicit probability estimates. A practical example of this common setting is clinicians often providing differential diagnoses as ranked conditions without numerical uncertainty. SoftEdit converts ranked answers into a tie-preserving pseudo-distribution (Appendix B.3), which is also used for compatible soft editors. As shown in Table 2, SoftEdit outperforms all soft editors in ranking alignment and generalization, measured by gated Kendall's τ (Appendix A.1.2). These results show that SoftEdit remains effective under weaker, ordinal feed-

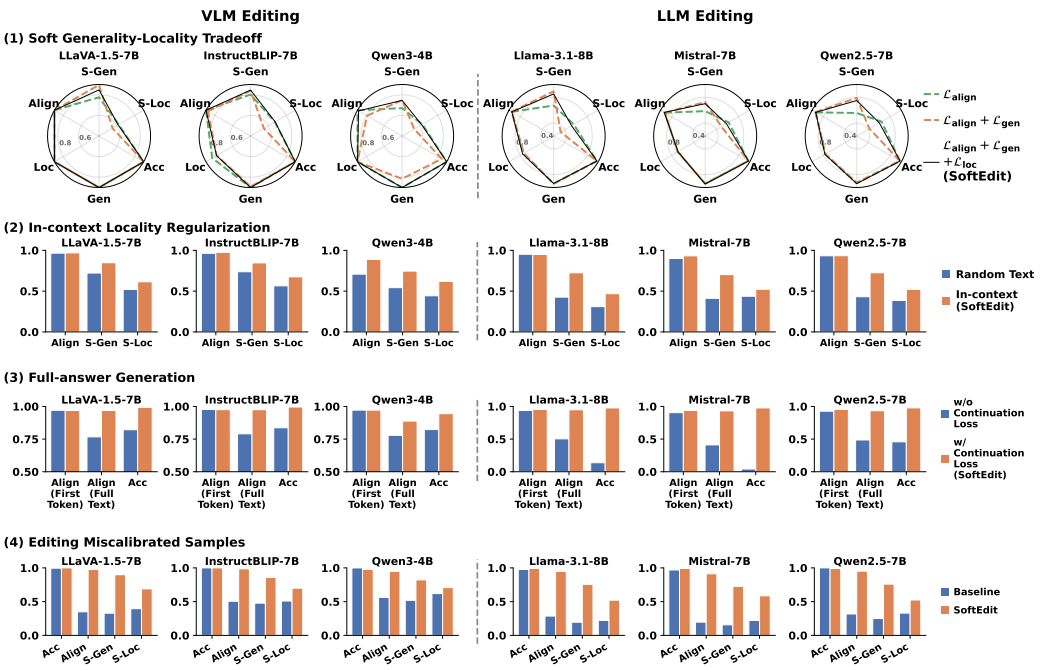

Figure 4: Ablation studies of SoftEdit by varying its objectives, locality samples, and input edits.

back: generalizing the pseudo-calibrated distribution generalizes the target ranking on related inputs, while locality preserves genuine rank changes in semantically similar inputs.

4.4 Ablation Studies

We next systematically ablate SoftEdit. Results are shown in Fig. 4.

(1) Soft generality–locality tradeoff. We first ablate SoftEdit by removing the distributional generality and locality terms, $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{loc}}$, one at a time. As shown in panel (1), optimizing $\mathcal{L}_{\text{align}}$ alone preserves real distribution shifts (high SLoc) but generalizes calibration poorly (low SGen), whereas adding $\mathcal{L}_{\text{gen}}$ alone improves SGen but decreases SLoc. By including both terms, SoftEdit optimizes both SGen and SLoc, effectively balancing the distributional generality and locality.

(2) In-context locality regularization. In panel (2), replacing answer-derived in-context locality text with random sentences substantially degrades calibration, showing that targeted local regularization is more effective than commonly used random locality samples.

(3) Full-answer continuation. In panel (3), we remove the continuation term from $\mathcal{L}_{\text{align}}$ and calibrate models using only first-token alignment. Without the continuation loss, the edited model fails to generate complete answers, leading to poor accuracy and answer-level distribution alignment.

(4) Correctly-answered but miscalibrated edits. In panel (4), we use 2,000 random samples from D_u where the original model generates answers correctly but remains distributionally miscalibrated. SoftEdit effectively calibrates these samples while preserving the original answer accuracy.

5 Conclusion

We introduce soft model editing, extending model editing from single-answer correction to targeted one-shot calibration using uncertainty in multi-answer human feedback. We propose SoftEdit, an uncertainty-aware editor that generalizes calibration to unseen related samples while preserving genuine distributional shifts. Across LLMs and VLMs, SoftEdit achieves state-of-the-art performance and efficiency, particularly for real-time sequential editing. Interesting future directions include detecting labeling errors from aleatoric uncertainty and extending soft editing to multi-output settings, such as differential diagnosis tasks for medical language models.

AI USE STATEMENT

In this work, we used generative AI tools for polishing of writing. We also used generative AI tools to generate synthetic data and clean and reformat datasets. GPT-4o was used to generate semantically equivalent question paraphrases, image captions, answer-conditioned textual context, and to assist in validating generated images as described in A.2.1. FLUX.2-klein-9B was used to generate and edit synthetic image variants for the evaluation of generality and locality, as described in A.2.1. We also used GPT-4o to clean the VQAv2 dataset by unifying synonymous free-form answer labels, as described in C.1. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, or interpret results. Assistance with translation is not applicable to this work. We have reviewed all AI-assisted work and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Anonymous code is included in the supplemental material. All code needed to reproduce our findings will be publicly available on GitHub. All data will be publicly available on Hugging Face.

REFERENCES

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7 (11), 2006.

Michael Boratko, Xiang Lorraine Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. Protoqa: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1122–1136, 2020.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. Can we edit multimodal large language models? *arXiv preprint arXiv:2310.08475*, 2023.

Zelei Cheng, Xian Wu, Jiahao Yu, Shuo Han, Xin-Qiang Cai, and Xinyu Xing. Soft-label integration for robust toxicity classification. *Advances in Neural Information Processing Systems*, 37:94776–94807, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, volume 2025, pp. 16366–16396, 2025.

Eve Fleisig, Su Lin Blodgett, Dan Klein, and Zeerak Talat. The perspectivist paradigm shift: Assumptions and challenges of capturing human labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2279–2292, 2024.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jia-Chen Gu, Hao-Xiang Xu, Jun-Yu Ma, Pan Lu, Zhen-Hua Ling, Kai-Wei Chang, and Nanyun Peng. Model editing can hurt general abilities of large language models. *CoRR*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Dongliang Guo, Mengxuan Hu, Zihan Guan, Thomas Hartvigsen, and Sheng Li. Balancedit: Dynamically balancing the generality-locality trade-off in multi-modal model editing. *arXiv preprint arXiv:2505.01343*, 2025.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. *Advances in Neural Information Processing Systems*, 36:47934–47959, 2023.

Warren He, Bo Li, and Dawn Song. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. Simbench: Benchmarking the ability of large language models to simulate human behaviors. In *International Conference on Learning Representations*, volume 2026, pp. 28290–28341, 2026.

Han Huang, Haitian Zhong, Tao Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Vlkeb: A large vision-language model knowledge editing benchmark. *Advances in Neural Information Processing Systems*, 37:9257–9280, 2024.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

Keito Inoshita, Xiaokang Zhou, Akira Kawai, and Katsutoshi Yada. Llms capture emotion labels, not emotion uncertainty: Distributional analysis and calibration of human-llm judgment gaps. *arXiv preprint arXiv:2604.27345*, 2026.

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, 2023. URL https://arxiv.org/abs/2310.06825.

Yanru Jiang and Siyu Liang. Capturing epistemic uncertainty in llm-based soft labeling. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 177–190, 2026.

Tom Joy, Francesco Pinto, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14919–14926, 2023.

Kyu-Hwan Jung. Large language models in medicine: clinical applications, technical challenges, and ethical considerations. *Healthcare Informatics Research*, 31(2):114–124, 2025.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Noah Lee, Na Min An, and James Thorne. Can large language models capture dissenting human voices? In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4569–4585, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296. IEEE, 2024.

Rachel Luo, Aadyot Bhatnagar, Yu Bai, Shengjia Zhao, Huan Wang, Caiming Xiong, Silvio Savarese, Stefano Ermon, Edward Schmerling, and Marco Pavone. Local calibration: metrics and recalibration. In *Uncertainty in Artificial Intelligence*, pp. 1286–1295. PMLR, 2022.

Marcus Ma, Georgios Chochlakis, Niyantha Maruthu Pandiyan, Jesse Thomason, and Shrikanth Narayanan. Large language models do multi-label classification differently. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2472–2495, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Takahiro Mimori, Keiko Sasada, Hirotaka Matsui, and Issei Sato. Diagnostic uncertainty calibration: Towards reliable machine predictions in medical domain. In *International Conference on Artificial Intelligence and Statistics*, pp. 3664–3672. PMLR, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Jingwei Ni, Yu Fan, Vilém Zouhar, Donya Rooein, Alexander Miserlis Hoyle, Mrinmaya Sachan, Markus Leippold, Dirk Hovy, and Elliott Ash. Can reasoning help large language models capture human annotator disagreement? In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–54, 2026.

Tim Pearce, Alexandra Brintrup, and Jun Zhu. Understanding softmax confidence and uncertainty. *arXiv preprint arXiv:2106.04972*, 2021.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9617–9626, 2019.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Jiaxing Qiu, Kaihua Hou, Roxana Daneshjou, Ahmed Alaa, and Thomas Hartvigsen. Reasonedit: Editing vision-language models using human reasoning. *arXiv preprint arXiv:2602.02408*, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. *arXiv preprint arXiv:2004.00345*, 2020.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 819–862, 2022.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37, 2024.

Jingyao Wu, Ashley Wang, Keane Ong, Paul Pu Liang, and Rosalind Picard. Shala-llm: Smartly handling ambiguous labels in aligning llms. *arXiv preprint arXiv:2606.05376*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pp. 2001, 2001.

Zhen Zeng, Leijiang Gu, Xun Yang, Zhangling Duan, Zenglin Shi, and Meng Wang. Visual-oriented fine-grained knowledge editing for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2491–2500, 2025.

Junzhe Zhang, Huixuan Zhang, Xunjian Yin, Baizhou Huang, Xu Zhang, Xinyu Hu, and Xiaojun Wan. Mc-mke: A fine-grained multimodal knowledge editing benchmark emphasizing modality consistency. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 17430–17445, 2025.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4862–4876, 2023.

A Evaluation Details

Let $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$ denote an edit, where x is an input query to a pretrained model, $\mathcal{Y}_K = (y_1, \ldots, y_K)$ is the corresponding set of K plausible human-provided answers. For a language model, x is a text question; for a vision-language model, x consists of an image and a text question. The K answers follow an answer-level target distribution $\mathbf{p}_K(\cdot \mid x) = (p_1, \ldots, p_K)$ estimated from human feedback, where p_k denotes the probability that a human provides y_k as the answer.

We aim to develop a model editor $g(\cdot)$ that takes in an edit e and a pretrained model f, and outputs a locally calibrated model $f_{\text{new}} = g(e, f)$ that achieves the following properties: Given the edit query x, let $\widehat{\mathbf{p}}_K(\cdot \mid x; f_{\text{new}})$ denote the model's distribution over the K human-provided answers under stochastic generation. f_{new} should align $\widehat{\mathbf{p}}_K$ with the human target distribution $\mathbf{p}_K$, such that each answer y_k is generated in full text with target probability p_k. Further, f_{new} should generalize the calibrated distribution to unseen queries that are semantically and distributionally similar to the edit, while preserving genuine distributional shifts in other unseen queries.

Evaluating f_{new} requires comparing its answer-level distribution $\widehat{\mathbf{p}}_K$ against the human target $\mathbf{p}_K$. Because Monte Carlo estimation of $\widehat{\mathbf{p}}_K$ requires expensive sampling and decoding full responses per query, we first introduce a fast, deterministic readout that approximates the answer-level distribution using a single forward pass for first-token probability combined with teacher-forced continuation validation for text generation (Sec. A.1). Building on this readout, we define two alignment metrics: the *gated Jensen–Shannon distance* $\mathrm{JS}_{\mathrm{gate}}$, which measures distributional closeness between $\widehat{\mathbf{p}}_K$ and $\mathbf{p}_K$ (Sec. A.1.1), and *gated Kendall's* τ_{gate}, which measures rank-order agreement between the top-K model-generated answers and the K human-provided answers (Sec. A.1.2).

We then detail the evaluation procedure for the edited model f_{new} in Sec. A.2 from two complementary perspectives. First, we evaluate the editor's conventional ***one-shot model editing*** performance using accuracy-based metrics—whether f_{new} generates a correct answer from $\mathcal{Y}_K^H$ on the edit input (*reliability*) and on semantically equivalent inputs (*generality*), and whether it preserves correct predictions on unrelated inputs (*locality*)—and describe the corresponding dataset construction in Sec. A.2.2. Second, we evaluate the editor's ***one-shot calibration*** performance on the model—whether f_{new} produces a well-calibrated answer distribution on the edit input (*alignment*), generalizes this calibration to semantically and distributionally similar inputs (*soft generality*), and preserves genuine distributional shifts (*soft locality*)—and describe the construction of evaluation data for each property in Sec. A.2.1.

Lastly, we provide details of the LLM-as-a-judge validation of generated images used for VLM editing evaluation in Sec. A.3.

A.1 Fast Answer-Level Distribution Estimation without Monte Carlo Sampling

We define a model's *answer-level* distribution by the probability that a stochastic generation from the model would produce each answer y_k as a complete text response. The standard approach is Monte Carlo estimation Lee et al. (2023); Sorensen et al. (2022): sample N full generations $\{\tilde{y}^{(1)}, \ldots, \tilde{y}^{(N)}\}$ and estimate the answer-level probability by the empirical frequency of exact text matches, $p_{\mathrm{MC}}(y_k \mid x) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\big[\tilde{y}^{(n)} = y_k\big]$. However, this requires sampling and decoding N full responses per query, and is prohibitively expensive when the model is evaluated at scale.

A common alternative is to use the model's first-token probability as a direct proxy for the answer-level probability Lee et al. (2023); Ma et al. (2025). This requires only a single forward pass, but the first-token probability is only an upper bound on the probability of the complete answer: it does not verify whether the model would continue along that branch to generate the intended multi-token answer, and multiple answers may share the same first token.

We therefore introduce a fast answer-level probability estimation method by combining the first-token distribution with a teacher-forced continuation gate to validate complete answer generation.

Setup. Let the human answers be

$$\mathcal{Y}_K = \{y_1, \ldots, y_K\}, \qquad y_k = (t_{k,1}, t_{k,2}, \ldots, t_{k,L_k}),$$

with human first-token set

$$\mathcal{T} = \{t_{1,1}, \ldots, t_{K,1}\}.$$

Let the model readout space be

$$\widehat{\mathcal{Y}} = \{\widehat{y}_1, \ldots, \widehat{y}_K, \widehat{y}_{K+1}\}, \qquad \widehat{y}_{K+1} \equiv \text{OTHER},$$

where $\widehat{y}_1, \ldots, \widehat{y}_K$ are the model's top-K most probable first-token answers, with corresponding first tokens

$$\widehat{\mathcal{T}} = \{\widehat{t}_{1,1}, \ldots, \widehat{t}_{K,1}\},$$

and exact first-step probabilities $\widehat{p}(\widehat{t}_{j,1} \mid x)$. The OTHER category collects all remaining probability mass not assigned to the top-K answers.

Continuation gate. For each model answer $\widehat{y}_j$ whose first token matches a human answer's first token, $\widehat{t}_{j,1} = t_{k,1} \in \mathcal{T}$, we check whether the model would complete the full answer y_k under teacher forcing accuracy:

$$c_j(x) = \mathbf{1}\big[\text{TFAcc}(t_{k,2:L_k} \mid x,\, t_{k,1}) = 1\big].$$

Specifically, the prompt x and the full answer tokens $(t_{k,1}, \ldots, t_{k,L_k})$ are concatenated and fed to the model in a single forward pass. At each continuation position $\ell = 2, \ldots, L_k$, the model's most likely prediction is $\widehat{t}_{k,\ell} = \arg\max \widehat{p}(\cdot \mid x, t_{k,1}, \ldots, t_{k,\ell-1})$. If $\widehat{t}_{k,\ell} = t_{k,\ell}$ for every position ℓ, that is, if the argmax prediction matches the target token at every continuation position, the gate passes and $c_j = 1$; otherwise $c_j = 0$.

Gated readout. For $j = 1, \ldots, K$, we estimate answer-level probability as

$$\widehat{p}(\widehat{y}_j \mid x) = \begin{cases} p_\theta(\widehat{t}_{j,1} \mid x) \cdot c_j(x), & \widehat{t}_{j,1} = t_{k,1} \in \mathcal{T}, \\ p_\theta(\widehat{t}_{j,1} \mid x), & \widehat{t}_{j,1} \notin \mathcal{T}. \end{cases}$$

That is, if a model answer's first token matches a human answer's first token and the continuation validation gate passes ($c_j = 1$), the first-token probability is retained as the answer-level estimate. If the continuation validation fails ($c_j = 0$), the probability is zeroed because the model fails to completely generate the answer.

All probability mass not ultimately assigned to the K human-provided answers is collected into a residual category OTHER. This includes both model answers whose first token does not match any human answer ($\widehat{t}_{j,1} \notin \mathcal{T}$) and any mass zeroed by a failed continuation gate:

$$\widehat{p}(\widehat{y}_{K+1} \mid x) = \widehat{p}(\text{OTHER} \mid x) = 1 - \sum_{j=1}^{K} \widehat{p}(\widehat{y}_j \mid x).$$

Hence, the complete readout of the model's probability over the K human-provided answers is the $(K+1)$-dimensional vector $\widehat{\mathbf{p}}(\cdot \mid x) = \big[\widehat{p}(\widehat{y}_1 \mid x), \ldots, \widehat{p}(\widehat{y}_K \mid x),\ \widehat{p}(\text{OTHER} \mid x)\big]$.

Discussion. This approach requires only one forward pass for the full-vocabulary first-token distribution plus at most K teacher-forced continuation checks, providing a fast approximation to the Monte Carlo answer-level distribution. It improves upon a first-token-only probability readout Lee et al. (2023); Ma et al. (2025) in two ways: (1) the continuation gate validates that the model would complete the full answer, zeroing the probability when it would not; (2) probability mass is retained in a residual OTHER category rather than renormalized via softmax over the human answer set, so model predictions outside the human answer set are penalized explicitly.

We note an inherent limitation shared by first-token-based readout: when multiple human answers share the same first token (e.g., "red car" and "red bus"), the readout cannot distinguish them at the first-token level. In this case, we sum the probabilities of answers that share the same first token into a single probability score for that first token in both the human-target and model distributions, so that the distributional alignment metrics (e.g., JS_{gate}) correctly compare the probability associated with each first token. Then, the continuation gate checks each colliding answer, and the gate passes if any of them passes, validating that the model would complete at least one of the colliding answers.

A.1.1 Distributional Alignment: Gated Jensen–Shannon Distance $\mathrm{JS}_{\mathrm{gate}}$

We measure distributional closeness between the human target $\mathbf{p}_K(\cdot \mid x)$ and the model gated readout $\widehat{\mathbf{p}}_K(\cdot \mid x)$ using the Jensen–Shannon (JS) distance. Both distributions are first projected onto the shared $(K+1)$-dimensional readout space $\widehat{\mathcal{Y}} = \{\widehat{y}_1, \ldots, \widehat{y}_K, \mathrm{OTHER}\}$ defined above. For the human-provided target answers, probabilities sharing the same first token are summed, and $p_k(\mathrm{OTHER} \mid x) = 0$. For the model, the gated readout $\widehat{\mathbf{p}}(\cdot \mid x)$ is defined as above. The JS divergence is

$$\mathrm{JSD}\big(\widehat{\mathbf{p}}_K \| \mathbf{p}_K\big) = \frac{1}{2} \sum_{j=1}^{K+1} \left[\widehat{p}_j \log_2 \frac{\widehat{p}_j}{m_j} + p_j \log_2 \frac{p_j}{m_j} \right], \qquad m_j = \tfrac{1}{2}\big(\widehat{p}_j + p_j\big),$$

We define the gated JS distance (lower is better) based on the gated readout as

$$\mathrm{JS}_{\mathrm{gate}} = \sqrt{\mathrm{JSD}\big(\widehat{\mathbf{p}}_K \| \mathbf{p}_K\big)} \ \in \ [0, 1]. \tag{1}$$

A.1.2 Ranking Alignment: Gated Kendall's τ_{gate}

When only the ranking of the K human answers is available (without explicit probability estimates), we measure rank-order agreement using Kendall's τ adapted for ties.

Both the human and model sides are represented as *tier-lists*: ordered groups of answers, best first, where answers within the same group are tied. On the human side, answers with equal vote counts form a tier and are mapped to first-token keys. On the model side, gated readout probabilities $\widehat{p}(\widehat{y}_j \mid x)$ with equal values form a tier (the OTHER entry is excluded). We take the union of all human and model first-token keys as the item set $\mathcal{U}$; any item missing from one side is placed in a worst-rank tier on that side.

We then compare every pair of items $(a, b) \in \mathcal{U}$. A pair is *concordant* if the human and model sides agree—either both rank a above b (or vice versa), or both tie them. A pair is *discordant* in all other cases: the two sides reverse the order, or one side ties them while the other does not. The metric is the fraction of concordant pairs:

$$\tau_{\mathrm{gate}} = \frac{|\{\text{concordant pairs}\}|}{\binom{|\mathcal{U}|}{2}} \ \in \ [0, 1]. \tag{2}$$

Higher is better. Every pair is weighted equally, making τ_{gate} an ordinal complement to $\mathrm{JS}_{\mathrm{gate}}$.

A.2 Evaluation Metrics

We evaluate the editor's performance from two perspectives. First, the editor should perform the new one-shot local calibration on the pretrained model, thus we evaluate performances based on distributional alignment—whether f_{new} produces a well-calibrated answer distribution on the edit input (*alignment*), generalizes this calibration to semantically equivalent inputs (*soft generality*), and preserves the model's sensitivity to genuine distributional shifts (*soft locality*) (Sec. A.2.1). Second, as a model editor, the one-shot calibration-enabled editor should still be able to perform conventional model editing, thus we evaluate performances based on answer generation accuracy—whether f_{new} generates a correct answer on the edit input (*reliability*) and on semantically equivalent inputs (*generality*), and whether it preserves correct predictions on unrelated inputs (*locality*) (Sec. A.2.2).

A.2.1 One-shot Model Calibration Performances

Given a pretrained model's predictions on a multi-answer dataset, the edit set D_{edit} contains errors made by the model ($f(x) \notin \mathcal{Y}_K$); these are the wrong model generations to be edited. The remaining samples Du, on which the model was already correct ($f(x) \in \mathcal{Y}_K$), are the unrelated samples used to preserve the model's original predictions.

Alignment (Align). For each edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$ in D_{edit}, the edited model should align its answer-level distribution with the human target distribution $\mathbf{p}_K$. We measure alignment using one

minus the gated Jensen–Shannon distance $\mathrm{JS}_{\mathrm{gate}}$ (Eq. 1) between the human target distribution and the edited model's gated readout:

$$\mathrm{Align} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\mathrm{gate}}(\mathbf{p}_K(\cdot \mid x), \widehat{\mathbf{p}}_K(\cdot \mid x;\, f_{\text{new}})) \right].$$

Higher is better, with $\mathrm{Align} = 1$ indicating perfect distributional alignment.

Soft Generality (SGen). For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let $R(x)$ denote semantically similar variants of the input x that preserve the target distribution $\mathbf{p}_K$. The edited model should preserve the calibrated answer distribution on these related inputs:

$$\mathrm{SGen} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\mathrm{gate}}(\mathbf{p}_K(\cdot \mid x), \widehat{\mathbf{p}}_K(\cdot \mid R(x);\, f_{\text{new}})) \right].$$

Higher is better, with $\mathrm{SGen} = 1$ indicating perfect generalization of the target distribution.

- ***Text Soft Generality (SGen-T).***
  For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let x_t denote the text question in x. We evaluate whether the edited model follows the calibrated answer distribution for semantically equivalent rephrasings $R(x_t)$:

  $$\text{SGen-T} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\mathrm{gate}}(\mathbf{p}_K(\cdot \mid x), \widehat{\mathbf{p}}_K(\cdot \mid R(x_t);\, f_{\text{new}})) \right].$$

  For each edit, we use GPT-4o to generate 4 rephrased questions with the prompt following prior works (Guo et al., 2025; Qiu et al., 2026): `Rephrase the following question in 4 different ways while keeping the meaning. Question: question.` Each rephrased question is used directly (for LLMs) or paired with the original image (for VLMs). Because the semantic meaning is preserved under rephrasing, the original target distribution is also preserved, i.e., $\mathbf{p}_K(\cdot \mid R(x_t)) = \mathbf{p}_K(\cdot \mid x)$.
  Examples of SGen-T samples are provided in Table 3.
- ***Image Soft Generality (SGen-I, VLMs only).***
  For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let x_i denote the image and x_t its associated text question. We evaluate whether the edited model preserves the calibrated answer distribution for semantically and distributionally close image variants $R(x_i)$ while keeping the original text question unchanged:

  $$\text{SGen-I} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \mathrm{JS}_{\mathrm{gate}}(\mathbf{p}_K(\cdot \mid x), \widehat{\mathbf{p}}_K(\cdot \mid R(x_i), x_t;\, f_{\text{new}})) \right].$$

  For each edit, we construct a set of distribution-preserving image variants in two ways:
  (1) Photometric perturbations (2 images). We apply a joint combination of Gaussian noise, brightness and contrast jitter, aspect-ratio stretching, and rotation to the original image. Exposed regions from stretching and rotation are filled with a randomly sampled real background scene from VQAv2 (Goyal et al., 2017). These photometric augmentations preserve the semantic content of the original image and are therefore associated with the same target distribution $\mathbf{p}_K$ as the edit.
  (2) Generated semantically and distributionally similar variants. To evaluate generalization to visually novel but semantically and distributionally similar images, we generate new images that approximate the edit's target distribution $\mathbf{p}_K$, following the generated-image generality evaluation setting of prior works (Cheng et al., 2023; Guo et al., 2025; Qiu et al., 2026). For each edit, we use FLUX.2-klein-9B (Labs, 2025) to first generate a top-1 anchor image from a GPT-4o-generated caption of the original image, and then edit this anchor toward each of the top-K answers, producing answer-specific images $\mathrm{img}_j,\ j = 1, \ldots, K$ that share the same underlying visual context while depicting different answers. Let $\mathbf{z}_j$ denote the VAE latent embedding of img_j. We construct a target-distribution-weighted blend

  $$\mathbf{z}_{\text{blend}} = \sum_{j=1}^{K} p_j \, \mathbf{z}_j,$$

  and decode $\mathbf{z}_{\text{blend}}$ to obtain a visually novel image whose target distribution approximates the original edit distribution $\mathbf{p}_K$. To reduce approximation error from using a single blended image, we additionally interpolate from $\mathbf{z}_{\text{blend}}$ toward each answer-specific embedding $\mathbf{z}_j$ at mild strengths $t \in \{0.0, 0.3\}$, and average distributional alignment across the resulting variants. We repeat the entire procedure with two random seeds to obtain two visually distinct sets of generated variants. Examples of SGen-I samples are provided in Fig. 5.

Question	Target	SGen
Tell me something that's hard to do while driving in the snow.	(stop 0.51, turn 0.49)	What is something challenging to do when driving on snowy roads? Can you name a difficult task to perform while driving in snowy conditions? What's a hard thing to manage when operating a vehicle in the snow? What is one thing that's tricky to handle while driving in snowy weather?
Name a holiday song with a fictional character.	(rudolph 0.51, frosty the snowman 0.49)	What is a holiday song that features a fictional character? Can you identify a holiday song about a fictional character? Which holiday song includes a fictional character? Provide the name of a holiday song that mentions a fictional character.
Name a fruit that you can buy dried	(grape 0.34, banana 0.33, apricot 0.33)	What is a fruit that is available in dried form? Can you name a fruit that is sold dried? Which fruit can be purchased in a dried state? Identify a fruit that comes in a dried version.
Name a food that it's hard to keep your face clean while eating.	(ice cream 0.34, ribs 0.33, spaghetti 0.33)	What is a food that makes it difficult to keep your face clean while eating? Can you name a food that's challenging to eat without getting your face messy? What type of food is hard to eat without messing up your face? Identify a food that often causes your face to get dirty while eating.

Table 3: ProtoQA example edits and their text soft generality (SGen) evaluation samples.

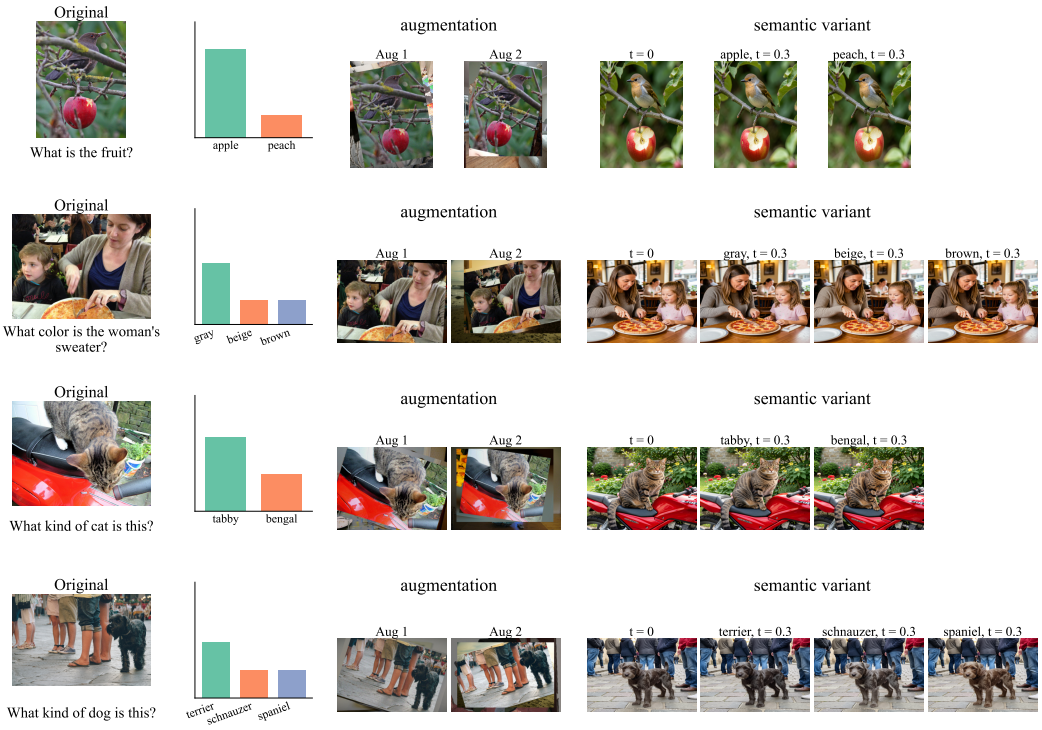

Figure 5: VQAv2 example edits and their SGen-I evaluation samples.

Soft Locality (SLoc). For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let $R_{\text{loc}}(x)$ denote semantically similar variants of x whose distributions over the target answers differ from the original edit's target distribution $\mathbf{p}_K$. The edited model should preserve these genuine distributional variations rather than over-generalizing the edit distribution to all related inputs. We measure soft locality as

$$\text{SLoc} = \mathbb{E}_{e \in D_{\text{edit}}} \left[1 - \text{JS}_{\text{gate}}(\mathbf{p}_K(\cdot \mid R_{\text{loc}}(x)), \widehat{\mathbf{p}}_K(\cdot \mid R_{\text{loc}}(x);\ f_{\text{new}})) \right].$$

Higher is better, with $\mathrm{SLoc} = 1$ indicating perfect preservation of the target distributional variations within the semantic neighborhood of an edit.

- ***Text Soft Locality (SLoc-T, LLMs only).***
  For an edit $e = (x_t, \mathcal{Y}_K, \mathbf{p}_K)$, we construct a semantically related variant x_k for each target answer $y_k \in \mathcal{Y}_K$ by adding textual context that unambiguously favors y_k. The edited model should therefore shift its answer distribution from the original multi-answer target $\mathbf{p}_K$ to the one-hot target $\mathbf{e}_k$. We measure text soft locality as

$$\text{SLoc-T} = \mathbb{E}_{e \in D_{\text{edit}}} \left[\frac{1}{K} \sum_{k=1}^{K} \left(1 - \mathrm{JS}_{\text{gate}}(\mathbf{e}_k, \widehat{\mathbf{p}}_K(\cdot \mid x_k;\, f_{\text{new}}))\right) \right].$$

  For each edit, we construct the distributional text locality samples by adding short textual context to the original question. Specifically, for each edit $e = (x_t, \mathcal{Y}_K, \mathbf{p}_K)$ and each target answer $y_k \in \mathcal{Y}_K$, we use GPT-4o to generate a brief premise c_k that makes y_k unambiguously the most plausible answer when prepended to the question, without explicitly mentioning the answer itself. The generation prompt follows the general form:

```
Given the question, target answer, and other plausible answers,
write a brief natural premise that makes the target answer
clearly the most likely answer without directly revealing it.
```

  The resulting input $x_k = c_k \oplus x_t$ is assigned the one-hot target distribution $\mathbf{e}_k$. Thus, each premise induces a controlled distributional shift from the original multi-answer distribution $\mathbf{p}_K$ toward a single target answer y_k. Unlike the image modality, where we construct intermediate shifts at multiple strengths, the text field uses a single fully disambiguated variant for each answer. We generate one such variant for each of the K target answers.
  Examples of SLoc-T samples are provided in Table 4.

- ***Image Soft Locality (SLoc-I, VLMs only).***
  For a VLM edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let $\mathrm{img}_{k,t}$ denote an image variant whose visual evidence is shifted toward answer y_k at strength t. We assign it the target distribution

$$\mathbf{p}_{k,t} = (1-t)\mathbf{p}_K + t\mathbf{e}_k,$$

  where $\mathbf{e}_k$ is the one-hot distribution on y_k. We measure image soft locality by how well the edited model follows these controlled distributional shifts:

$$\text{SLoc-I} = \mathbb{E}_{e \in D_{\text{edit}}} \left[\frac{\sum_{k=1}^{K} \sum_{t \in \mathcal{T}} t \left[1 - \mathrm{JS}_{\text{gate}}\big(\mathbf{p}_{k,t}, \widehat{\mathbf{p}}_K(\cdot \mid \mathrm{img}_{k,t}, x_t;\, f_{\text{new}})\big)\right]}{K \sum_{t \in \mathcal{T}} t} \right].$$

  Higher is better, with larger distributional shifts receiving greater weight.
  We construct the image variants $\mathrm{img}_{k,t}$ in two complementary ways.
  (1) *Editing the original image toward each answer.* We first construct distributionally shifted variants while preserving the original scene and visual context. For each edit, we use FLUX.2-klein-9B (Labs, 2025) to edit the original image toward each target answer y_k at full strength (t=1), producing an unambiguous image img_k. We use GPT-4o to verify that each target label is visually recognizable and discard samples for which any edit fails. Let $\mathbf{z}_{\text{orig}}$ and $\mathbf{z}_k$ denote the VAE latent embeddings of the original image and the full-strength edit toward y_k. We generate intermediate variants by

$$\mathbf{z}_{k,t} = (1-t)\mathbf{z}_{\text{orig}} + t\mathbf{z}_k, \qquad t \in \{0.1,\, 0.3,\, 0.5,\, 0.7,\, 1.0\},$$

  and decode $\mathbf{z}_{k,t}$ to obtain $\mathrm{img}_{k,t}$. Thus, increasing t progressively shifts the image from the original ambiguity toward an unambiguous depiction of y_k.
  (2) *Editing generated semantically similar images toward each answer.* We additionally construct the same distributional shifts on visually novel but semantically related images. For each edit, we first use FLUX.2-klein-9B to generate a top-1 anchor image from a GPT-4o-generated caption of the original image, and then edit this anchor toward each of the top-K answers, producing answer-specific images $\mathrm{img}_j,\ j = 1, \ldots, K$ that share the same underlying visual structure. Let $\mathbf{z}_j$ denote their latent embeddings. We compute a target-distribution-weighted blend

$$\mathbf{z}_{\text{blend}} = \sum_{j=1}^{K} p_j \mathbf{z}_j,$$

Question	Target	SLoc: Answer-specific premises (italic) prepended to question
Tell me something that's hard to do while driving in the snow.	(stop 0.51, turn 0.49)	
	stop	*In snowy conditions, vehicles often lose traction, making it difficult to avoid sliding forward when attempting to slow down.* Tell me something that's hard to do while driving in the snow.
	turn	*When roads are icy, maintaining control during directional changes becomes particularly challenging.* Tell me something that's hard to do while driving in the snow.
Name a holiday song with a fictional character.	(rudolph 0.51, frosty the snowman 0.49)	
	rudolph	*This song tells the story of a reindeer who helps guide Santa's sleigh on a foggy Christmas Eve.* Name a holiday song with a fictional character.
	frosty the snowman	*This song features a magical figure who comes to life with the help of a special hat.* Name a holiday song with a fictional character.
Name a fruit that you can buy dried	(grape 0.34, banana 0.33, apricot 0.33)	
	grape	*This fruit is commonly dried to make one of the most popular snack items worldwide.* Name a fruit that you can buy dried
	banana	*This fruit is commonly sliced into thin pieces and dried into chips, often used as a snack or topping.* Name a fruit that you can buy dried
	apricot	*This fruit is commonly dried and retains a sweet, tangy flavor, often used in trail mixes and cooking.* Name a fruit that you can buy dried
Name a food that it's hard to keep your face clean while eating.	(ice cream 0.34, ribs 0.33, spaghetti 0.33)	
	ice cream	*This food is often eaten on a cone and tends to melt quickly, especially in warm weather.* Name a food that it's hard to keep your face clean while eating.
	ribs	*This food is often coated in a sticky sauce that gets all over your hands and face while eating.* Name a food that it's hard to keep your face clean while eating.
	spaghetti	*This dish is known for its long strands often coated in a saucy topping that can easily splatter.* Name a food that it's hard to keep your face clean while eating.

Table 4: ProtoQA example edits and their text soft locality (SLoc) evaluation samples.

which serves as a proxy for the original ambiguity under $\mathbf{p}_K$. For each answer y_k, we interpolate

$$\mathbf{z}_{k,t} = (1-t)\mathbf{z}_{\text{blend}} + t\mathbf{z}_k, \qquad t \in \{0.0,\ 0.3,\ 0.5,\ 0.7,\ 1.0\},$$

and decode $\mathbf{z}_{k,t}$ to obtain the corresponding variants. We repeat the procedure with two random seeds to obtain two visually distinct sets of trajectories.

Examples of SLoc-I samples are provided in Fig 6.

A.2.2 One-shot Model Editing Performances

Reliability (Acc). The edited model should output a correct answer from the target answers for each edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$ in D_{edit}:

$$\text{Acc} = \mathbb{E}_{e \in D_{\text{edit}}}\, \mathbf{1}\big\{ f_{\text{new}}(x) \in \mathcal{Y}_K \big\}.$$

Locality (Loc). The edited model should preserve its original correct prediction on unrelated samples D_u:

$$\text{Loc} = \mathbb{E}_{x \in D_u}\, \mathbf{1}\big\{ f_{\text{new}}(x) = f(x) \big\}.$$

Generality (Gen). For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let $R(x)$ denote semantically similar variants of the input x for which the edited model should generate the same target answers $\mathcal{Y}_K$ according to the target distribution $\mathbf{p}_K$. The edited model should output a correct answer from the target set:

$$\text{Gen} = \mathbb{E}_{e \in D_{\text{edit}}} \mathbf{1}\big\{ f_{\text{new}}(R(x)) \in \mathcal{Y}_K \big\}.$$

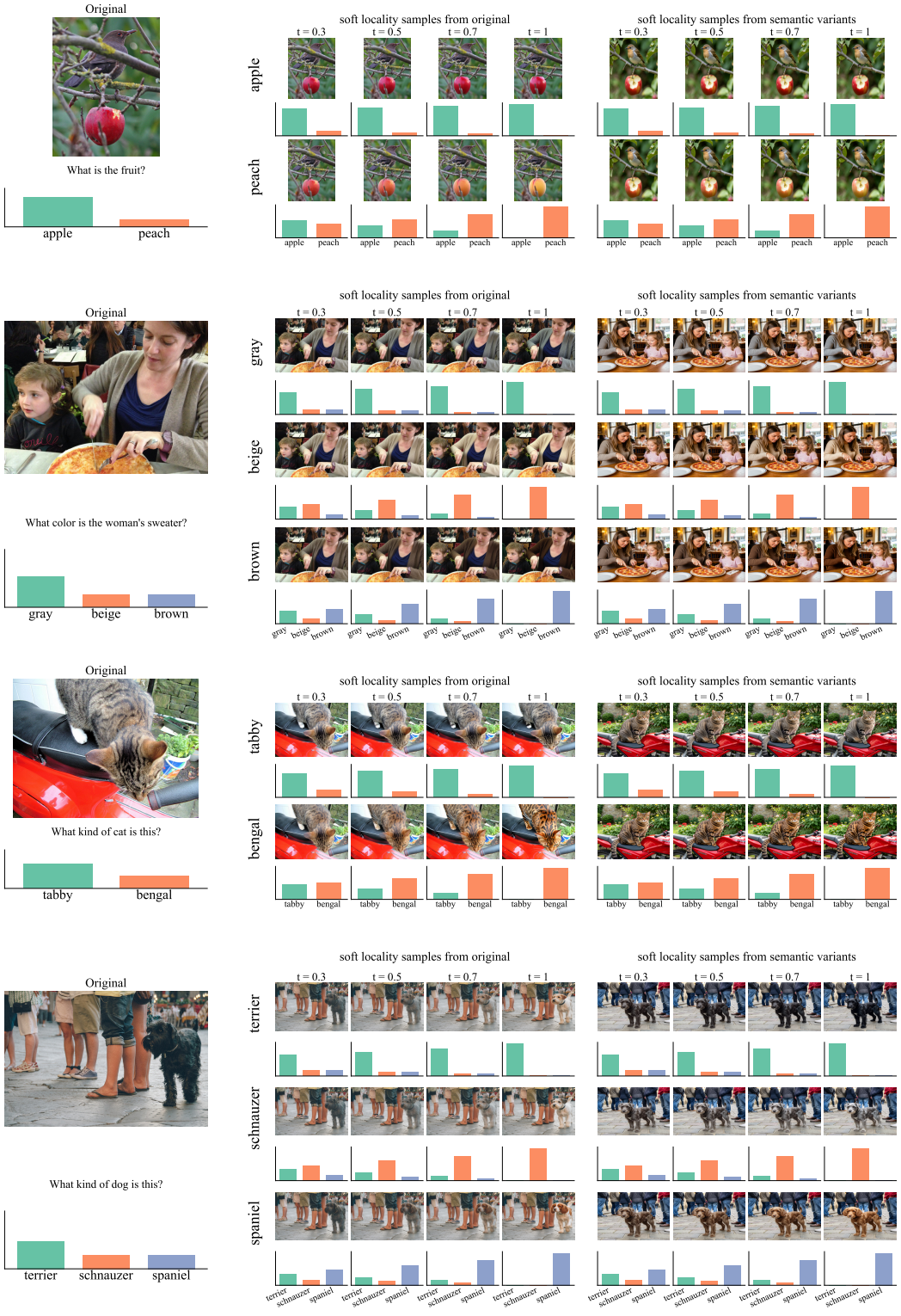

Figure 6: VQAv2 example edits and their SLoc-I evaluation samples.

- ***Text generality (Gen-T).*** For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let x_t denote a text question in x. We evaluate whether the edited model generates a correct answer for rephrased questions $R(x_t)$:

$$\text{Gen-T} = \mathbb{E}_{e \in D_{\text{edit}}} \mathbf{1}\big\{ f_{\text{new}}(R(x_t)) \in \mathcal{Y}_K \big\}.$$

  We use the same rephrased questions $R(x_t)$ constructed for SGen-T, but evaluate answer-generation accuracy rather than distributional alignment.
- ***Image generality (Gen-I, VLMs only).*** For an edit $e = (x, \mathcal{Y}_K, \mathbf{p}_K)$, let x_i denote an image and x_t denote its associated text question in the input x. We evaluate whether the edited model generates a correct answer for related semantic-preserving images $R(x_i)$ that correspond to the same target answers $\mathcal{Y}_K$ following $\mathbf{p}_K$, while keeping the original text question unchanged:

$$\text{Gen-I} = \mathbb{E}_{e \in D_{\text{edit}}} \mathbf{1}\big\{ f_{\text{new}}(R(x_i), x_t) \in \mathcal{Y}_K \big\}.$$

  We use the same image variants $R(x_i)$ constructed for SGen-I, but evaluate answer-generation accuracy rather than distributional alignment.

A.3 Validation of Generated Images

We follow prior work Cheng et al. (2023); Zhang et al. (2025); Guo et al. (2025); Qiu et al. (2026) to generate images for VLM-editing evaluation metrics. We validate every generated answer-specific image with GPT-4o, before interpolating answer-specific images to obtain intermediate-strength evaluation images. Specifically, for each image we pose two binary questions to GPT-4o: (1) whether the image depicts the intended answer label (*e.g.*, "Does this image show a {label}?"), and (2) whether the image contains any stamped or watermark-style text. An image is marked as failed if the target label is not depicted in the image, or if the answer label is leaked as watermark text onto the image. We repeat this generation-and-validation cycle up to three times with different random seeds, and discard samples that have any image that still fails after three attempts. For VQAv2, out of 78,872 generated answer-specific images, 16,992 failed the content check (target label not shown) and 12,207 failed the watermark check (answer text leaked onto the image). After discarding all samples with at least one failed image, a total of 19,186 VQAv2 samples are retained with valid evaluation images for the soft generality (SGen-I) and soft locality (SLoc-I) metrics. This constitutes a sufficiently large evaluation set for benchmarking editing performance across all VLM editing methods.

B SoftEdit Details

B.1 Automatic Loss Balancing

Both λ_{loc} and λ_{gen} are set automatically from a pilot set of N_{pilot} edits evaluated at $\Delta W_e = 0$ (a single forward pass per edit, no optimization).

λ_{loc}: Equalizing the regularization terms. The natural scales of $\mathcal{L}_{\text{gen}}$ and $\mathcal{L}_{\text{loc}}$ differ across models and datasets. To bring them to comparable magnitude, we compute the pilot-set means:

$$\bar{\mathcal{L}}_{\text{gen}} = \frac{1}{N_{\text{pilot}}} \sum_{n=1}^{N_{\text{pilot}}} \mathcal{L}_{\text{gen}}^{(n)}, \qquad \bar{\mathcal{L}}_{\text{loc}} = \frac{1}{N_{\text{pilot}}} \sum_{n=1}^{N_{\text{pilot}}} \mathcal{L}_{\text{loc}}^{(n)},$$

and set λ_{loc} to their ratio, rounded to the nearest half-decade grid point:

$$\lambda_{\text{loc}} = \text{round}\left(\frac{\bar{\mathcal{L}}_{\text{gen}}}{\bar{\mathcal{L}}_{\text{loc}}} \right).$$

This ensures that $\lambda_{\text{loc}} \cdot \mathcal{L}_{\text{loc}}$ is approximately the same scale as $\mathcal{L}_{\text{gen}}$, so the two regularization terms contribute equally regardless of model or dataset.

λ_{gen}: Scaling regularization relative to alignment. λ_{gen} controls the overall weight of the combined regularization block $\lambda_{\text{gen}}(\mathcal{L}_{\text{gen}} + \lambda_{\text{loc}}\, \mathcal{L}_{\text{loc}})$ relative to the alignment term $\mathcal{L}_{\text{align}}$. The alignment term is the primary objective and should dominate; λ_{gen} ensures this while allowing the regularization to be promoted when alignment is already well-scaled.

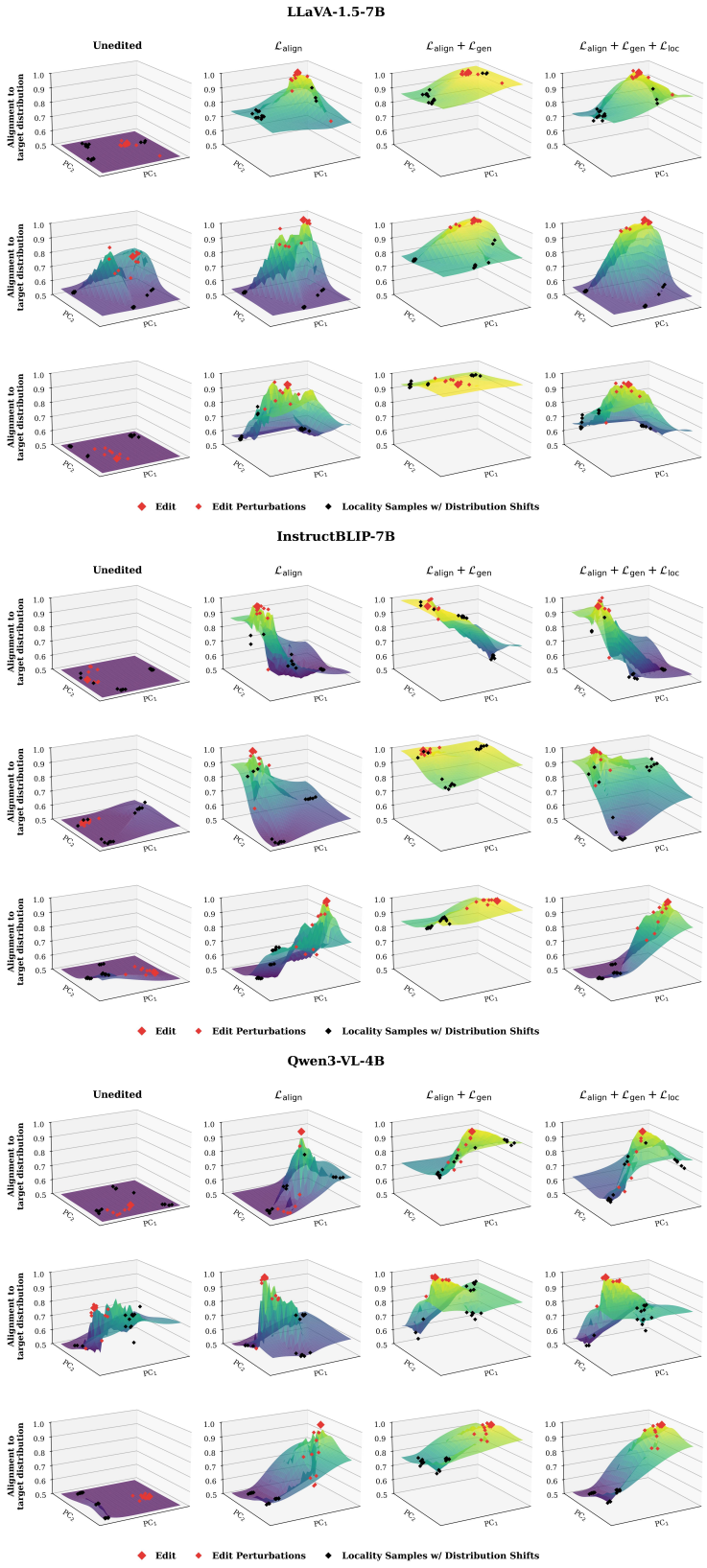

Figure 7: Visualization of changes in the local distribution field of more example edits from VQAv2, when editing three VLMs.

Using the same pilot set, we also compute:

$$\bar{\mathcal{L}}_{\text{align}} = \frac{1}{N_{\text{pilot}}} \sum_{n=1}^{N_{\text{pilot}}} \mathcal{L}_{\text{align}}^{(n)}.$$

We set λ_{gen} so that the alignment loss is approximately ρ times larger than the total regularization:

$$\lambda_{\text{gen}} = \text{clip}\left(\text{round}\left(\frac{\bar{\mathcal{L}}_{\text{align}}}{\rho \cdot (\bar{\mathcal{L}}_{\text{gen}} + \lambda_{\text{loc}}\, \bar{\mathcal{L}}_{\text{loc}})}\right),\ 1,\ 10\right),$$

where ρ is a fixed target dominance ratio (we use $\rho = 2$ in all experiments). The $\text{round}(\cdot)$ function snaps to the same half-decade grid as above. The $\text{clip}(\cdot,\ 1,\ 10)$ clamp has the following effect: Lower bound ($\lambda_{\text{gen}} \geq 1$): If alignment is already much larger than the regularization (ratio $> \rho$), λ_{gen} boosts regularization toward the target ratio. If regularization is already comparable to or larger than alignment, λ_{gen} stays at 1—it never suppresses regularization below its natural scale. Upper bound ($\lambda_{\text{gen}} \leq 10$): Caps the boost to prevent regularization from overwhelming alignment in extreme cases. Both λ_{loc} and λ_{gen} are computed once from the pilot set and held fixed for all edits. The entire procedure adds negligible cost (one forward pass per pilot edit with no gradient computation).

B.2 Automatic Retrieval Rejection Threshold

Prior retrieval-based editors (Qiu et al., 2026) reject out-of-distribution queries using a fixed distance percentile computed from the pairwise distances among codebook keys. This percentile must be hand-tuned per model and dataset, and its effective meaning shifts as the codebook grows. We address this with two improvements: a distance threshold estimated from the codebook's own geometry and automatically calibrated, and a paraphrase judge for semantic verification.

Estimating the semantic proximity scale from the codebook. The codebook entries are an empirical sample of edit queries from the deployment distribution, where most edits concern different topics. The nearest-neighbor distances among these entries therefore provide a natural estimate of the distance scale at which "semantically related" ends and "coincidentally nearest" begins.

Given N codebook key embeddings $\{k_1, \ldots, k_N\}$, we compute each entry's nearest-neighbor distance (excluding itself):

$$d_i = \min_{j \neq i} \|k_i - k_j\|_2, \qquad i = 1, \ldots, N.$$

The distribution $\{d_1, \ldots, d_N\}$ characterizes the typical inter-edit spacing in the embedding space. A new query with embedding k_q is matched to its nearest entry k_{j^*} and rejected if $\|k_q - k_{j^*}\|_2$ exceeds a threshold τ set as a percentile of this distribution:

$$\tau = \text{percentile}_p(d_1, \ldots, d_N).$$

Intuitively, a query that is farther from its nearest edit than most edits are from each other is unlikely to be semantically related. Prior retrieval-based editors (Qiu et al., 2026) hand-tune this percentile per model and dataset; we instead calibrate it automatically.

Auto-tuning the percentile. A lower percentile yields a tighter gate (better locality but risk of missing valid paraphrases); a higher one is more permissive (better generality but risk of false retrievals). After the first batch of edits, we generate known-related perturbations for a sample of codebook entries by paraphrasing the question and augmenting the image, then encode each perturbation into the key space. We sweep candidate percentiles and measure retrieval accuracy: whether each perturbation is correctly matched back to its parent edit (passing both the distance gate and the paraphrase judge). We select the tightest percentile whose accuracy does not drop by more than a tolerance δ relative to the most lenient gate:

$$p^* = \min\left\{p : \text{hits}(p) \geq \text{hits}(p_{\max}) - \delta\right\},$$

where $p_{\max}$ is the loosest candidate and $\delta = 5$ in all experiments. This adapts the gate to the model's own embedding geometry without manual tuning.

Paraphrase judge. Pure distance gating cannot distinguish queries that are close in embedding space but ask about different entities. As a second filter, a lightweight LLM-based paraphrase judge (Qwen2.5-1.5B-Instruct) verifies whether the query and the matched edit's question ask about the same entity. Given the two questions, the judge produces a Yes/No probability; the query is accepted only if $P(\text{Yes}) \geq 0.5$.

B.3 One-Shot Calibration with Ranked Answers

SoftEdit can also be used when human feedback provides only a *ranking* of plausible answers rather than explicit probability estimates. Such cases reflect realistic scenarios—for example, a clinician may list differential diagnoses in order of likelihood without assigning numeric probabilities to each.

To apply SoftEdit in this setting, we convert the ranked answers into a pseudo-distribution and use it in place of the annotator-frequency target $\mathbf{p}_K$; the SoftEdit objective and all its components ($\mathcal{L}_{\text{align}}$, $\mathcal{L}_{\text{gen}}$, $\mathcal{L}_{\text{loc}}$) remain unchanged.

From rankings to a pseudo-distribution. We represent the human ranking as an *ordered partition* of the K answers into M rank groups that preserves ties:

$$\mathcal{R} = [R_1, R_2, \ldots, R_M],$$

where R_1 contains the most plausible answers, and answers within the same group R_r are tied (unordered). To construct the pseudo-distribution, we first assign an exponential weight to each rank group using a temperature $\tau > 0$:

$$w_r = \frac{\exp\big[-(r-1)/\tau\big]}{\sum_{s=1}^{M} \exp\big[-(s-1)/\tau\big]},$$

so that higher-ranked groups receive more mass and the sharpness is controlled by τ. We then split each group's mass equally among its members. The pseudo-probability of answer $y \in R_r$ is:

$$\tilde{p}(y) = \frac{w_r}{|R_r|}.$$

This construction ensures that (i) tied answers always receive equal probability, and (ii) a larger tie group does *not* receive more total mass—rank r always owns w_r regardless of $|R_r|$. At the extremes, $\tau \to \infty$ yields a uniform distribution over rank groups (then split within ties), and $\tau \to 0$ concentrates all mass on the top group R_1. We use $\tilde{p}$ as the target distribution $\mathbf{p}_K$ and apply SoftEdit without modification. Evaluation replaces distributional alignment (JS distance) with ranking alignment (Kendall's τ, Appendix A.1.2), an ordinal metric that measures pairwise rank concordance between the model's predicted ranking and the human ranking, with proper handling of ties.

C Experimental Details

C.1 Datasets

VQAv2. VQAv2 (Goyal et al., 2017) is a large-scale visual question answering benchmark built on MS-COCO images, where each question is independently answered by 10 crowd-sourced annotators. We aggregate the 10 free-form answers per question into a vote-count distribution over unique normalized labels, then apply a multi-stage filter to retain only samples that exhibit genuine multi-annotator uncertainty. Specifically, we (i) unify synonymous free-form labels via GPT-4o (*e.g.* "no one" → "nobody"), (ii) drop samples with non-alphabetic or multi-word labels, and yes/no-only polls, (iii) merge hyponym–hypernym pairs and remove residual ambiguous samples, and (iv) retain only single-target questions. *To reduce label noise and preserve samples containing high aleatoric uncertainty* due to genuine ambiguity or subjective disagreement, we drop any singleton labels (count = 1), and keep only samples in the top 50th percentile of vote-distribution empirical entropy. After all preprocessing stages, the final VQAv2 benchmark dataset comprises 19,186 multi-annotator samples—a large-scale evaluation set with sufficient coverage to benchmark sequential model editing across thousands of edits per backbone.

ProtoQA. ProtoQA (Boratko et al., 2020) is a commonsense question-answering dataset, where human respondents provide free-form answers that are clustered into canonical answer groups with associated vote counts—yielding a distribution over plausible natural language answers per question. We load both the train and validation splits from HuggingFace and normalize each cluster's canonical answer to lowercase. *To retain meaningful distributional disagreement among respondents and reduce label noise*, we keep only the top-3 answer clusters by vote count per question as the gold distribution and discard questions with fewer than two surviving clusters after this truncation. The resulting benchmark dataset contains 9,762 multi-answer commonsense questions, providing a large pool for evaluating editing methods on LLMs.

C.2 Backbone VLMs and LLMs

Edited VLMs. LLaVA-1.5-7B (Liu et al., 2024) connects a CLIP ViT-L/14 vision encoder to a Vicuna-7B language model through a two-layer MLP projector and is trained using a two-stage visual instruction-tuning pipeline. InstructBLIP-7B (Dai et al., 2023) is an instruction-tuned BLIP-2 variant that employs a frozen image encoder and an instruction-aware Q-Former to extract visual features conditioned on the input instruction, which are projected into a frozen Vicuna-7B language model for generation. Qwen3-VL-4B-Instruct (Yang et al., 2025) employs a Vision Transformer for variable-resolution image encoding and integrates multi-level visual features into a 4B-parameter transformer decoder through DeepStack, with Interleaved-MRoPE for unified spatial–temporal positional modeling.

Edited LLMs. Llama-3.1-8B-Instruct (Grattafiori et al., 2024) is an 8B-parameter decoder-only transformer with grouped query attention and rotary position embeddings, supporting a 128K-token context window and post-trained for instruction following. Mistral-7B-Instruct (Jiang et al., 2023) is a 7B-parameter decoder-only transformer that employs sliding window attention and grouped query attention for efficient long-context inference, with an instruction-tuned variant for instruction following. Qwen2.5-7B-Instruct (Qwen et al., 2025) is a 7B-parameter decoder-only transformer with grouped query attention, rotary position embeddings, and SwiGLU activations, post-trained for instruction following and general-purpose language understanding and generation.

C.3 Baseline Model Editing Methods

We compare SoftEdit against state-of-the-art model editing methods adapted for one-shot calibration of pretrained large language models. Three baseline editing methods—Fine-tuning, MEND, and GRACE—have been adapted to edit both LLMs and VLMs. We additionally include the state-of-the-art VLM editors BalancEdit and ReasonEdit, and the text-only editors IKE and AlphaEdit for LLMs. All baselines are adapted to a *soft-label* variant that uses the multi-answer distribution instead of a single target answer to supervise the editing, ensuring each method has access to the same distributional target as SoftEdit. Specifically, (1) Fine-tuning (FT) has been adopted to update the weight of a selected layer of the pretrained model on each edit via gradient descent. It is the simplest weight-editing baseline but is prone to catastrophic collapse as edits accumulate. (2) MEND (Mitchell et al., 2021) trains a lightweight meta-network that decomposes and transforms the raw fine-tuning gradient into a low-rank parameter update, enabling fast edits without per-edit iterative optimization. (3) GRACE (Hartvigsen et al., 2023) maintains a discrete key–value codebook alongside the frozen model, where each edit stores a hidden activation as a key and learns a corresponding replacement activation as its value; at inference time, the value is retrieved only when an input falls within the associated local influence radius. (4) In-context Knowledge Editing (IKE)(Zheng et al., 2023; Guo et al., 2025) is a retrieval-based, training-free editor that represents edits as textual demonstrations and retrieves relevant edits at inference time to prepend to the input context, guiding the model toward the updated knowledge without modifying its parameters. (5) BalancEdit (Guo et al., 2025) extends localized codebook editing to multimodal models by generating positive and negative samples for each edit to estimate its influence scope, which is encoded into a discrete latent-space codebook without modifying the underlying model weights. (6) ReasonEdit (Qiu et al., 2026) stores human-provided reasoning in a codebook and uses a topology-balanced multimodal embedding method to retrieve relevant reasoning at inference time to guide generation. (7) AlphaEdit (Fang et al., 2025) is a locate-then-edit method that constrains param-

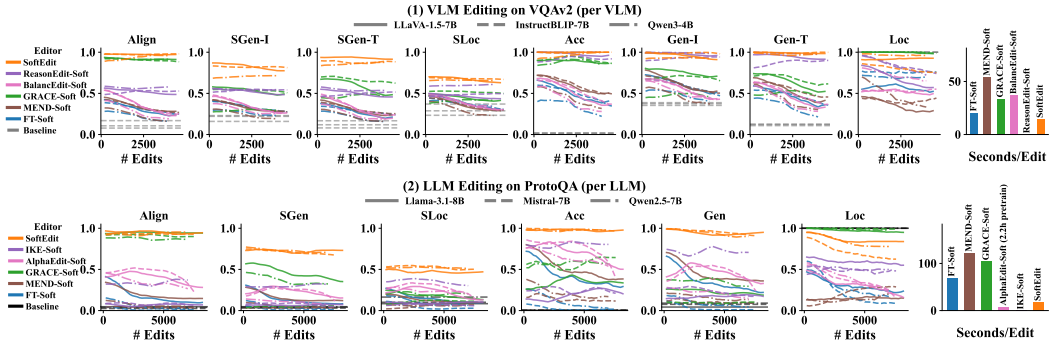

Figure 8: Sequential editing performance and computational efficiency across VLM and LLM editors adapted to one-shot calibration. SoftEdit achieves the best alignment, soft generality, and soft locality. Each line corresponds to an edited pretrained models.

eter updates by projecting them onto the null space of preserved-knowledge activations, reducing interference with previously stored knowledge during sequential editing.

D LIMITATIONS

SoftEdit has limitations. First, SoftEdit focuses on *aleatoric uncertainty*: irreducible uncertainty arising from genuine ambiguity or subjective disagreement, where the observed human answer distribution is treated as reliable feedback. In practice, however, human feedback may also contain annotation noise, unreliable answers, or outliers that should not be interpreted as meaningful uncertainty. Distinguishing these cases is an important direction for future work. For example, properties of the observed answer distribution, such as its shape and entropy, could potentially be combined with outlier detection or annotator-consistency signals to identify unreliable answers or edits before calibration. Second, SoftEdit inherits the scalability limitations of retrieval-based editors. Although each edit stores only a compact weight perturbation matrix, the retrieval index and stored perturbations grow with the number of edits, while retrieval introduces additional inference-time overhead. These costs may become non-negligible under large-scale sequential editing. Future work could improve scalability through more efficient GPU-aware retrieval and storage, compression or quantization of stored perturbations, and mechanisms for pruning obsolete or unreliable entries.