https://doi.org/10.1038/s44387-026-00114-1

An information-theoretic framework for robust large language model editing

Qizhou Chen[1,2], Chengyu Wang[2], Taolin Zhang[3] & Xiaofeng He[1] ✉

Large Language Models (LLMs) have become indispensable tools in science, technology, and society, enabling transformative advances across diverse fields. However, errors or outdated information within these models can undermine their accuracy and restrict their safe deployment. Developing efficient strategies for updating model knowledge without the expense and disruption of full retraining remains a critical challenge. Current model editing techniques frequently struggle to generalize corrections beyond narrow domains, leading to unintended consequences and limiting their practical impact. Here, we introduce a novel framework for editing LLMs, grounded in information bottleneck theory. This approach precisely compresses and isolates the essential information required for generalizable knowledge correction while minimizing disruption to unrelated model behaviors. Building upon this foundation, we present the Information Bottleneck Knowledge Editor (IBKE), which leverages compact latent representations to guide gradient-based updates, enabling robust and broadly applicable model editing. We validate IBKE's effectiveness across multiple LLM architectures and standard benchmark tasks, demonstrating state-of-the-art accuracy, as well as improved generality and specificity of edits. These findings establish a theoretically principled and practical paradigm for open-domain knowledge editing, advancing the utility and trustworthiness of LLMs in real-world applications.

Large language models (LLMs)[1–3] have revolutionized artificial intelligence, rapidly transforming fields such as finance[4–6], medicine[7–10], and education[11–13]. Their remarkable ability to process and generate human-like text has made them indispensable across a wide range of applications. However, as LLMs are increasingly employed in complex, dynamic environments, inherited outdated or erroneous information from training data presents significant challenges[14–16], particularly in high-risk and knowledge-intensive sectors. Efficiently updating the internal knowledge of LLMs without complete retraining has therefore become a crucial research priority. Model editing techniques make it possible to introduce targeted corrections, enabling models to integrate new knowledge or rectify errors while reducing computational demands and mitigating the risk of catastrophic forgetting[17,18].

Recent investigations into the explainability of knowledge localization within transformer models have spurred significant advances in knowledge editing[19–21]. Contemporary approaches typically focus on identifying and directly modifying localized, knowledge-sensitive regions in the network, a process termed "locate-then-edit" (L&E)[21–24]. Alternatively, methods such as transformer patching[25] bypass localization by optimizing a new neuron for each edit. Despite their promise, both paradigms face notable limitations: optimizing a single piece of knowledge can often result in overfitting, leading to updates that are highly confident yet narrow and that may fail to generalize to related queries or concepts[26]. As a consequence, performance on complex questions, especially those requiring multi-step reasoning or the handling of diverse inputs, is compromised[27,28]. As illustrated in Fig. 1, reliance on a single edit prior restricts the domain of edit generalization, making it narrow and difficult to control.

Recent work has sought to enhance the edit prior by incorporating hypernetworks with edit training[17,29–32]. Meanwhile, datasets[27,28] have been developed to broaden the scope of edit domains and increase diversity in edit generalization. Nonetheless, overfitting continues to pose a challenge: model editors often excel on familiar data but struggle with out-of-domain knowledge, limiting their practical utility. As shown in Fig. 1, edit training-based methods typically perform well in Domain A, where the prior is dominant, but their performance diminishes in Domain B.

To overcome these challenges, we propose a model editing framework grounded in the information bottleneck (IB) principle[33–35]. By constraining the flow of information during knowledge updates, the IB approach enables editors to extract and preserve features most relevant for generalization while omitting redundant or irrelevant details. Conceptually, our method formulates model editing as a two-stage process: first, the edit request is transformed into a latent representation capturing the essential

[1]East China Normal University, Shanghai, China. [2]Alibaba Group, Hangzhou, China. [3]Hefei University of Technology, Hefei, China. ✉e-mail: hexf@cs.ecnu.edu.cn

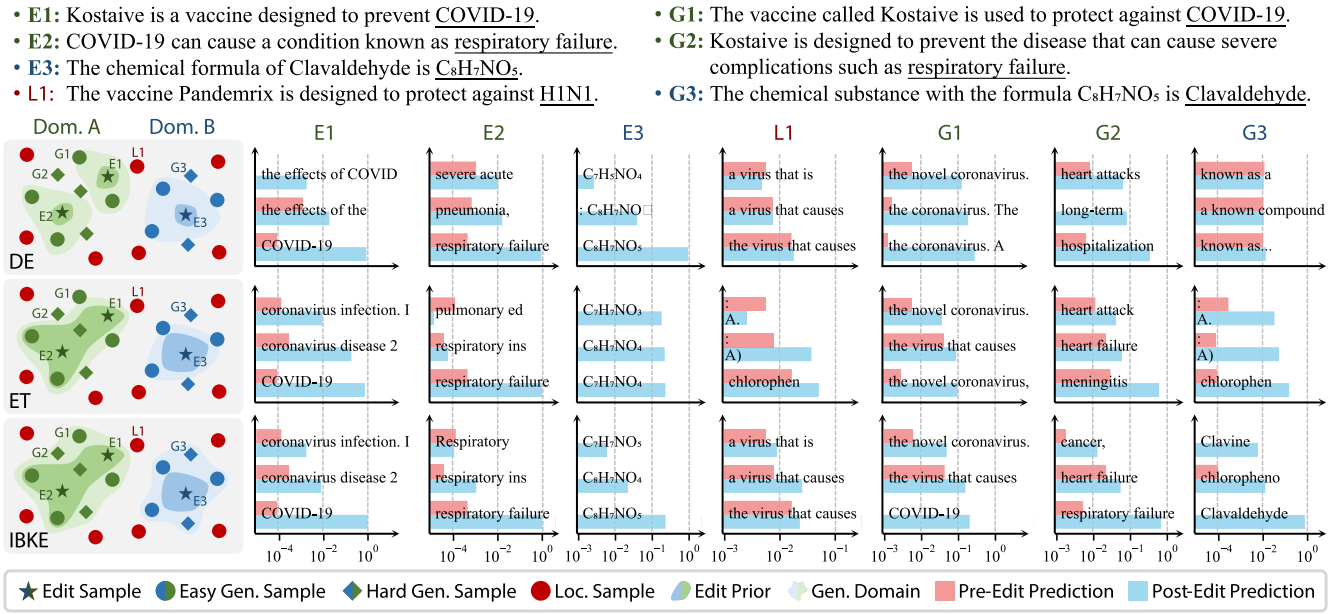


Fig. 1 | Comparison of two model editing paradigms with IBKE. Direct Editing (DE) is based solely on the edit sample as its prior, while Edit Training (ET) leverages richer priors; when combined with the IB, ET achieves improved generalization. The bar chart presents the top three prediction probabilities of the model after editing (using "E1", "E2", and "E3") for each corresponding sample. Results for the three editing paradigms are shown as follows: ROME (for DE), IBKE without IB, and IBKE with IB, all implemented using the Qwen3-1.7B backbone model. For illustration, Medicine (Domain A) and Chemistry (Domain B) are featured as example edit domains, with training performed on the Medicine domain. An effective model editor should correct the model's responses to generalization (Gen.) samples while preserving accuracy for locality (Loc.) samples.

information; second, this representation guides a targeted intervention. The intervention may take various forms, such as parameter adaptation or the integration of auxiliary modules to steer model responses. Our framework enforces three principles: minimizing transferred information to encourage compression, ensuring sufficiency for generalization so that edits propagate appropriately, and maintaining independence from unrelated knowledge to preserve locality. Together, these constraints promote concise and impactful updates that generalize well beyond a single corrected instance. As depicted in Fig. 1, applying the IB principle significantly expands the domain of edit generalization, even when the editing prior remains fixed.

Building on these theoretical foundations, we present the Information Bottleneck Knowledge Editor (IBKE). In its first stage, IBKE leverages first-order gradients with respect to the edit request[17,32], distilling these signals into a compact latent space that forms the information bottleneck. In the second stage, the resulting edit representation is used to calibrate gradients and rescale update strengths across different tokens. These enhanced gradients are then applied to update the model weights. Through edit training, IBKE incorporates training priors into the original gradients, thereby extending the domain of edit generalization beyond specific instances.

We validate the effectiveness of IBKE in extensive experiments across four knowledge editing datasets: ZSRE[36], CounterFact[21], MQuAKE[27], and UniEdit[28]. These evaluations span multiple large language model architectures, including GPT2-XL (1.5B), GPT-J (6B), Qwen3-1.5B, and Qwen3-8B. Our results show substantial improvements over existing methods, positioning IBKE as a robust and generalizable solution for open-domain model editing.

The remainder of this paper is organized as follows. The Results section first presents the empirical performance of IBKE across multiple datasets and analyzes its behavior under different settings. The Discussion section summarizes the main findings and discusses the limitations of this work as well as potential directions for future research. Finally, the Methods section provides the formal definition of the IB-based editing framework and the model architecture.

Results

Experimental Settings

For LLM backbones, we evaluated language model backbones with varying sizes and architectures, including GPT2-XL (1.5B) (https://huggingface.co/openai-community/gpt2-xl), GPT-J (6B) (https://huggingface.co/EleutherAI/gpt-j-6b), Qwen3-1.7B (https://huggingface.co/Qwen/Qwen3-1.7B), and Qwen3-8B (https://huggingface.co/Qwen/Qwen3-8B). All experiments employed full-precision (FP32) for GPT2-XL and Qwen3-1.7B, and half-precision (FP16) for GPT-J and Qwen3-8B.

For baseline editors, we categorized methods into two principal groups: Direct Editing (DE) and Edit Training (ET). Representative editors from both categories were selected to ensure coverage of diverse paradigms. DE methods include those that directly modify the model parameters, such as ROME[21], and AlphaEdit[24], as well as techniques employing external parameters or modules, including T-Patcher[25], LEMoE[37], and GRACE[38]. The ET category includes MEND[17], which utilizes a multilayer perceptron (MLP)-based hypernetwork to adjust model parameters, and SERAC[30], which incorporates a lightweight counterfactual model to generate edit-related responses. In addition, we include Fine-Tuning (FT)[14] as an additional baseline. Baseline hyperparameters were set in accordance with those reported in[39].

For the IBKE settings, building on prior research in knowledge localization, attribution analysis, and related editing techniques[21,22,39], we selected the output linear layer matrices from the feed-forward networks (FFNs) within editing-sensitive layers of each backbone model as the edit weights, denoted $\{W_i\}_{i=1}^{n}$. Specifically, we applied edits to layers 15, 16, and 17 in GPT2-XL; layers 6, 7, and 8 in GPT-J; and layers 18, 19, and 20 in both Qwen3 variants. Consistent with the approach described in[17], we set the dimension of the hypernetwork module to $d_m = 1920$. Based on hyperparameter searching, the sequence length parameter was set to $l_m = 10$, and the IB regularization coefficient to $\beta = 0.1$. For edit training, we used a learning rate of 1×10^{-4} and a maximum batch size of 8, with batch size adjusted dynamically according to the interactions among edit samples.

Table 1 | Statistics of each dataset, where CF, MH, OD, and CC denote counterfactual, multi-hop, open-domain, and combinations of diverse criteria, respectively

Datasets	Train Size	Test Size	CF	MH	OD	CC
ZSRE	163,196	19,086				
CounterFact	10,000	10,000	✓			
MQuAKE	2,036	10,182	✓	✓		
UniEdit	207,419	103,723		✓	✓	✓

To ensure a rigorous evaluation, datasets were selected according to criteria such as the complexity of edit generalization, dataset scale, and the inclusion of counterfactual knowledge. Specifically, we employ four widely used datasets: ZSRE[17,36], CounterFact[21], MQuAKE[27], and UniEdit[28]. ZSRE is a question answering (QA) dataset in which generalization samples are created by back-translation, resulting in paraphrased questions that form equivalence neighborhoods. CounterFact is a more challenging dataset consisting of counterfactual statements that typically receive lower confidence scores from LLMs than their factual counterparts. MQuAKE is designed to evaluate whether edited models can successfully handle multi-hop reasoning questions related to the edited facts. UniEdit is a large-scale, open-domain dataset based on Wikidata[40], covering 25 human domains across five major sectors: Natural Sciences, Humanities, Social Sciences, Applied Sciences, and Interdisciplinary Studies. It utilizes a unified multi-hop triple-chain sampling algorithm to construct diverse and comprehensive evaluation scenarios. For all datasets except MQuAKE, we followed the original train–test splits. Because MQuAKE does not include a designated training set, we sampled one-sixth of its data for use in few-shot augmentation. Statistical details for each dataset are presented in Table 1.

For the training settings, IBKE, MEND, and SERAC were trained on the UniEdit training set. To assess the effect of small-scale data augmentation, we additionally included 512 samples from each of the training sets of CounterFact, ZSRE, and MQuAKE. We duplicated these samples to increase their proportion in the overall training data by tenfold to balance their training exposure with UniEdit. Results obtained with this augmentation are indicated with an asterisk (*). Early stopping was applied if validation loss ceased to decrease for 50,000 steps, and the total number of training steps was capped at 1,000,000. Model checkpoints are saved every 3,000 iterations, with the checkpoint achieving the lowest validation loss selected for final evaluation. Each training run required approximately 3–7 days on two NVIDIA A6000 GPUs (with a single GPU used for training GPT2-XL and Qwen3-1.7B).

For evaluation, following the procedure described in[28], we compute three metrics for model editing: reliability, generality, and locality. For the reliability score, we measure the top-1 token-level hit rate of the post-edited LLM on the designated edit targets. Generality is assessed by analyzing the predicted probability distribution over the target tokens and determining whether each token appears among the top five predictions (the top-5 hit rate), which reflects how effectively the post-edited LLM improves recall of the intended concepts. For locality, we first record the original predictions of the model on the locality samples and then evaluate the top-5 hit rate for these samples after model editing.

Evaluation of Overall Editing Performance

Table 2 summarizes the performance of various editing methods across a range of datasets and LLM backbones.

Direct Editing (DE)-based methods consistently achieve high reliability and locality scores. By localizing edits to sensitive model weights and employing low-rank adaptation, these approaches enforce strong edit confidence while minimizing disruption to unrelated knowledge. On the ZSRE dataset, where generalization primarily involves paraphrasing of edit samples, DE-based methods also perform strongly. As ZSRE is relatively straightforward, most methods achieve similarly high scores across the three metrics (within 3% of the top results). However, in counterfactual editing scenarios, such as those presented by CounterFact, these methods exhibit low generality, struggling to extend edits beyond direct instances. On more complex datasets, including UniEdit and MQuAKE, which feature diverse edit types, multi-hop reasoning, and interactions among multiple edits, DE-based approaches yield notably low generality scores. These findings indicate that DE-based methods tend to enhance prediction for directly edited instances without fostering broader integration of new knowledge within the LLM. Consequently, the model cannot reliably propagate edited knowledge to address indirect or ripple effects.

In contrast, Edit Training (ET)-based methods employ edit-specific training to establish interactions between newly introduced knowledge and the model's internal representations, leading to improved generality, as evidenced by their overall performance on UniEdit and ZSRE. However, because training is typically performed on real-world data, generality remains constrained for counterfactual datasets such as MQuAKE and CounterFact, which demand stronger intervention to override entrenched model priors. Notably, supplementing ET-based methods with small-sample data leads to substantial improvements in generality.

Specifically, IBKE* utilizes holistic representations of edit requests and integrates the IB mechanism to distill essential information, thereby achieving effective generalization while minimizing interference with unrelated knowledge. In comparison, MEND relies solely on an MLP for token-level gradient decomposition and lacks a unified approach to edit semantics; this often results in excessive editing of individual samples and diminished generalization. SERAC, similarly, is constrained by the capacity of its counterfactual sub-model and its single-retrieval design, which limits its ability to support complex interactions among multiple edits.

Figure 2 illustrates the trade-off between generality and locality, providing an intuitive evaluation of each editor's ability to calibrate LLM responses to edited inputs while preserving performance on unrelated queries. Scores are averaged over three discriminative datasets (excluding ZSRE). Analogous to recall and precision in classification tasks, the calculated *F1 score* (the harmonic mean of generality and locality) quantitatively reflects how precisely the editing method delineates the boundaries of knowledge modification. DE-based methods tend to cluster within or below the yellow band, indicating high locality but limited generality. ET-based methods achieve a more balanced performance, particularly after few-shot augmentation, with IBKE demonstrating the best overall trade-off.

For mass editing, similar to GRACE and SERAC, we employ a semantic retriever[41] for MEND and IBKE to perform coarse filtering, preventing numerical instability caused by excessive accumulation of weight updates[42–44]. Specifically, each edit sample is added to a retrieval pool, and for a given input prompt, the top-4 entries in the pool (the maximum hop in evaluation) with the highest semantic similarity are dynamically incorporated into the model to adapt its response. Table 3 presents the results after 3,000 edits, including MEMIT[22], a method specifically designed for mass editing, for comparison. Among DE-based methods, except for the retrieval-based GRACE, all other approaches experience varying degrees of performance degradation. GRACE decides whether to apply dynamic adaptation based on the linear distance between intermediate representations of the input prompt and edit statements. However, this strong semantic assumption makes it difficult for generality inputs to trigger the corresponding edits. SERAC trains an edit classifier to improve recall under semantic variations, but the small language model used to generate responses struggles to handle interactions among multiple edits, leading to similarly limited generality scores. In contrast, our simple threshold-free retrieval strategy largely avoids missed edits. Building upon this, IBKE demonstrates a stronger ability to balance interactions and conflicts among multiple edits.

The above results underscore the advantages of introducing edit priors via targeted edit training and incorporating the IB mechanism.

Hyperparameter Search

To optimize the performance of IBKE, we investigate the effects of two key hyperparameters: the learnable sequence length l_m and the IB balance

https://doi.org/10.1038/s44387-026-00114-1

Table 2 | Overall editing performance on UniEdit, MQuAKE, CounterFact, and ZSRE

			UniEdit				MQuAKE			CounterFact				ZSRE			
Backbones	Type	Editors	Rel.	Gen.	Loc.	Average	Rel.	Gen.	Average	Rel.	Gen.	Loc.	Average	Rel.	Gen.	Loc.	Average
		W/O	20.32	27.53	**100.00**	49.28$_{\pm 0.02}$	32.42	25.10	28.76$_{\pm 0.05}$	0.00	3.42	**100.00**	34.47$_{\pm 0.01}$	19.26	27.41	**100.00**	48.89$_{\pm 0.03}$
	Base	FT	99.10	60.43	89.68	83.07$_{\pm 0.56}$	99.80	44.08	71.94$_{\pm 0.23}$	99.25	58.34	77.56	78.38$_{\pm 0.36}$	99.47	96.75	97.89	98.04$_{\pm 0.17}$
		ROME	96.76	58.16	96.76	83.90$_{\pm 0.31}$	89.61	22.90	56.26$_{\pm 0.45}$	99.35	46.71	95.91	80.66$_{\pm 0.37}$	99.58	85.49	99.14	94.74$_{\pm 0.20}$
		T-Patcher	90.61	59.12	87.97	79.23$_{\pm 0.31}$	97.95	34.62	66.28$_{\pm 0.37}$	99.00	33.72	88.21	73.64$_{\pm 0.13}$	94.42	96.86	98.35	96.54$_{\pm 0.15}$
		LEMoE	78.87	60.18	89.61	76.22$_{\pm 0.24}$	72.68	41.80	57.24$_{\pm 0.52}$	**99.78**	47.45	93.20	80.14$_{\pm 0.03}$	97.50	96.44	99.03	97.66$_{\pm 0.12}$
		GRACE	**99.64**	46.58	98.75	81.66$_{\pm 0.09}$	**99.95**	24.99	62.47$_{\pm 0.04}$	99.57	3.77	98.59	67.31$_{\pm 0.15}$	99.11	27.52	99.80	75.48$_{\pm 0.13}$
	DE	AlphaEdit	97.01	59.00	96.67	84.23$_{\pm 0.42}$	95.21	24.69	59.95$_{\pm 0.40}$	99.12	62.58	98.58	86.76$_{\pm 0.31}$	99.10	93.27	99.48	97.29$_{\pm 0.25}$
		SERAC	98.59	78.39	87.36	88.11$_{\pm 0.37}$	83.91	20.08	51.99$_{\pm 0.11}$	77.17	28.94	82.12	62.75$_{\pm 0.18}$	92.58	83.04	99.64	91.75$_{\pm 0.18}$
		MEND	97.40	68.92	97.99	88.10$_{\pm 0.51}$	98.50	45.07	71.78$_{\pm 0.14}$	97.65	43.30	83.43	74.79$_{\pm 0.26}$	97.71	95.38	96.70	96.60$_{\pm 0.17}$
		IBKE	97.80	96.89	97.48	97.39$_{\pm 0.30}$	97.34	56.09	76.72$_{\pm 0.25}$	96.08	57.87	77.45	77.13$_{\pm 0.18}$	95.79	97.69	96.36	96.61$_{\pm 0.32}$
		SERAC*	98.93	81.82	87.33	89.36$_{\pm 0.16}$	99.38	47.71	73.54$_{\pm 0.13}$	99.49	83.78	88.56	90.61$_{\pm 0.16}$	**99.92**	**99.02**	99.85	**99.60**$_{\pm 0.11}$
		MEND*	97.22	69.64	97.19	88.02$_{\pm 0.28}$	95.98	79.48	87.73$_{\pm 0.03}$	97.73	76.69	89.18	87.87$_{\pm 0.06}$	98.24	98.47	99.04	98.58$_{\pm 0.06}$
GPT2-XL (1.5B)	ET	IBKE*	98.13	**97.00**	97.46	**97.53**$_{\pm 0.17}$	98.72	**92.31**	**95.51**$_{\pm 0.43}$	97.19	**88.59**	87.58	**91.12**$_{\pm 0.22}$	97.08	97.83	99.24	98.05$_{\pm 0.13}$
		W/O	24.90	32.63	**100.00**	52.51$_{\pm 0.05}$	35.24	26.43	30.84$_{\pm 0.04}$	0.44	2.59	**100.00**	34.34$_{\pm 0.05}$	21.56	29.92	**100.00**	50.50$_{\pm 0.03}$
	Base	FT	99.60	64.75	90.57	84.98$_{\pm 0.16}$	99.46	62.77	81.11$_{\pm 0.44}$	99.90	62.94	92.22	85.02$_{\pm 0.16}$	**100.00**	98.78	99.04	99.27$_{\pm 0.11}$
		ROME	97.26	64.68	95.81	85.92$_{\pm 0.32}$	88.63	24.82	56.73$_{\pm 0.35}$	99.20	83.82	93.91	92.31$_{\pm 0.31}$	99.27	97.79	**100.00**	99.02$_{\pm 0.07}$
		T-Patcher	90.81	44.73	89.42	74.99$_{\pm 0.23}$	89.34	14.03	51.69$_{\pm 0.16}$	89.87	31.92	87.83	69.87$_{\pm 0.31}$	93.16	92.10	94.26	93.17$_{\pm 0.38}$
		LEMoE	83.02	63.16	94.96	80.38$_{\pm 0.30}$	76.42	37.33	56.88$_{\pm 0.56}$	99.80	39.40	96.40	78.53$_{\pm 0.10}$	98.75	97.31	99.65	98.57$_{\pm 0.01}$
		GRACE	**99.88**	46.93	99.96	82.26$_{\pm 0.04}$	**99.85**	25.82	62.84$_{\pm 0.12}$	**100.00**	2.63	99.87	67.50$_{\pm 0.05}$	98.88	30.06	99.96	76.30$_{\pm 0.15}$
	DE	AlphaEdit	98.46	57.37	97.00	84.28$_{\pm 0.52}$	95.96	24.56	60.26$_{\pm 0.34}$	99.24	63.28	98.64	87.05$_{\pm 0.46}$	99.66	94.25	99.73	97.88$_{\pm 0.23}$
		SERAC	99.09	81.63	86.28	89.00$_{\pm 0.19}$	98.03	18.13	58.08$_{\pm 0.16}$	88.94	43.85	81.30	71.36$_{\pm 0.09}$	97.14	97.56	99.95	98.22$_{\pm 0.19}$
		MEND	97.21	68.01	96.23	87.15$_{\pm 0.09}$	97.19	47.15	72.17$_{\pm 0.23}$	97.50	41.86	84.18	74.51$_{\pm 0.20}$	98.47	96.30	94.65	96.47$_{\pm 0.28}$
		IBKE	98.01	**95.88**	96.34	**96.74**$_{\pm 0.06}$	96.03	56.32	76.18$_{\pm 0.20}$	96.51	58.30	78.73	77.85$_{\pm 0.17}$	96.25	98.81	98.74	97.93$_{\pm 0.34}$
		SERAC*	99.85	82.37	86.79	89.67$_{\pm 0.10}$	99.70	48.87	74.28$_{\pm 0.25}$	99.64	69.01	88.21	85.62$_{\pm 0.16}$	99.37	99.02	**100.00**	**99.46**$_{\pm 0.17}$
		MEND*	97.04	66.50	95.62	86.38$_{\pm 0.06}$	96.97	74.70	85.83$_{\pm 0.54}$	96.58	73.49	91.35	87.14$_{\pm 0.42}$	99.03	98.78	98.90	98.90$_{\pm 0.32}$
GPT-J (6B)	ET	IBKE*	97.76	94.89	95.54	96.06$_{\pm 0.33}$	99.05	**95.11**	**97.08**$_{\pm 0.16}$	97.18	**85.32**	94.62	**92.37**$_{\pm 0.34}$	98.81	**99.40**	99.14	99.12$_{\pm 0.16}$
		W/O	28.36	48.50	**100.00**	58.95$_{\pm 0.04}$	32.68	23.65	28.16$_{\pm 0.03}$	0.67	2.76	**100.00**	34.48$_{\pm 0.04}$	23.53	34.99	**100.00**	52.84$_{\pm 0.06}$
	Base	FT	92.95	69.03	68.36	76.78$_{\pm 0.17}$	99.42	35.36	67.39$_{\pm 0.30}$	99.95	86.25	40.77	75.66$_{\pm 0.30}$	**99.86**	**99.84**	95.48	98.39$_{\pm 0.04}$
		ROME	92.54	66.13	95.67	84.78$_{\pm 0.34}$	88.79	20.29	54.54$_{\pm 0.40}$	98.79	39.42	98.79	79.00$_{\pm 0.36}$	98.25	87.07	99.72	95.01$_{\pm 0.10}$
		T-Patcher	84.84	69.45	93.67	82.65$_{\pm 0.10}$	91.47	26.78	59.12$_{\pm 0.14}$	95.14	25.29	62.73	61.05$_{\pm 0.22}$	88.92	90.40	99.04	92.79$_{\pm 0.23}$
		LEMoE	79.77	69.36	92.08	80.40$_{\pm 0.10}$	62.74	35.11	48.93$_{\pm 0.49}$	99.87	50.65	96.55	82.36$_{\pm 0.05}$	98.91	99.15	99.77	99.27$_{\pm 0.06}$
		GRACE	98.67	68.79	99.99	89.15$_{\pm 0.11}$	**99.92**	23.80	61.86$_{\pm 0.08}$	**99.96**	2.77	99.99	67.58$_{\pm 0.13}$	98.73	35.13	99.98	77.95$_{\pm 0.08}$
	DE	AlphaEdit	93.87	64.92	98.95	85.91$_{\pm 0.17}$	95.30	24.13	59.71$_{\pm 0.59}$	99.03	37.31	99.82	78.72$_{\pm 0.22}$	95.79	80.25	99.59	91.88$_{\pm 0.22}$
		SERAC	99.23	82.87	86.46	89.52$_{\pm 0.04}$	99.53	11.78	55.66$_{\pm 0.33}$	93.54	26.32	88.11	69.32$_{\pm 0.30}$	97.43	91.16	99.84	96.14$_{\pm 0.13}$
		MEND	94.27	71.36	97.89	87.84$_{\pm 0.03}$	96.51	38.02	67.26$_{\pm 0.28}$	97.73	44.75	84.16	75.55$_{\pm 0.09}$	96.11	97.43	96.83	96.79$_{\pm 0.20}$
		IBKE	97.65	96.51	93.83	96.00$_{\pm 0.15}$	93.83	52.90	73.37$_{\pm 0.27}$	97.95	59.86	76.47	78.09$_{\pm 0.15}$	94.79	96.72	98.71	96.74$_{\pm 0.05}$
		SERAC*	**99.52**	84.53	88.63	90.89$_{\pm 0.10}$	99.69	45.70	72.69$_{\pm 0.11}$	99.87	85.58	91.93	92.46$_{\pm 0.05}$	99.61	98.00	99.94	99.18$_{\pm 0.14}$
		MEND*	94.65	72.09	97.14	87.96$_{\pm 0.45}$	97.12	74.90	86.01$_{\pm 0.34}$	96.51	70.48	91.39	86.12$_{\pm 0.06}$	98.05	98.10	98.54	98.23$_{\pm 0.03}$

Table 2 (continued) | Overall editing performance on UniEdit, MQuAKE, CounterFact, and ZSRE

Backbones	Type	Editors	UniEdit				MQuAKE			CounterFact				ZSRE			
			Rel.	Gen.	Loc.	Average	Rel.	Gen.	Average	Rel.	Gen.	Loc.	Average	Rel.	Gen.	Loc.	Average
Qwen3 (1.7B)	ET	IBKE*	97.72	**97.64**	95.27	**96.88**$_{\pm 0.26}$	97.54	**95.00**	**96.27**$_{\pm 0.27}$	98.66	**88.04**	93.75	**93.48**$_{\pm 0.22}$	99.20	99.55	99.52	**99.42**$_{\pm 0.21}$
		W/O	32.54	45.70	**100.00**	59.41$_{\pm 0.06}$	36.26	25.01	30.64$_{\pm 0.07}$	0.82	2.30	**100.00**	34.37$_{\pm 0.01}$	27.65	38.03	**100.00**	55.23$_{\pm 0.02}$
	Base	FT	99.83	68.90	96.39	88.37$_{\pm 0.06}$	99.99	32.07	66.03$_{\pm 0.07}$	99.85	32.29	88.89	73.68$_{\pm 0.08}$	99.73	95.44	99.64	98.27$_{\pm 0.19}$
		ROME	94.36	64.85	96.44	85.21$_{\pm 0.34}$	98.35	23.90	61.13$_{\pm 0.49}$	99.50	46.07	99.61	81.73$_{\pm 0.24}$	98.63	91.55	99.46	96.55$_{\pm 0.23}$
		T-Patcher	87.60	49.18	92.69	76.49$_{\pm 0.31}$	85.79	25.27	55.53$_{\pm 0.31}$	88.86	27.29	79.78	65.31$_{\pm 0.23}$	88.69	86.84	94.74	90.09$_{\pm 0.25}$
		LEMoE	81.74	63.87	95.76	80.45$_{\pm 0.27}$	68.79	34.64	51.71$_{\pm 0.17}$	99.88	42.65	98.20	80.24$_{\pm 0.05}$	**99.92**	**98.40**	99.94	**99.42**$_{\pm 0.09}$
		GRACE	**99.88**	64.05	99.99	87.97$_{\pm 0.10}$	**100.00**	25.93	62.96$_{\pm 0.08}$	**99.90**	2.29	99.99	67.39$_{\pm 0.14}$	99.36	38.01	99.95	79.11$_{\pm 0.04}$
	DE	AlphaEdit	95.74	64.16	99.12	86.34$_{\pm 0.09}$	91.55	23.51	57.53$_{\pm 0.58}$	99.52	44.96	99.81	81.43$_{\pm 0.28}$	97.95	85.30	99.61	94.29$_{\pm 0.24}$
		SERAC	98.24	81.79	83.50	87.84$_{\pm 0.21}$	98.69	14.65	56.67$_{\pm 0.16}$	89.76	41.63	82.51	71.30$_{\pm 0.10}$	94.25	89.87	99.76	94.63$_{\pm 0.20}$
		MEND	94.45	71.23	95.44	87.04$_{\pm 0.11}$	93.29	38.44	65.86$_{\pm 0.17}$	92.98	45.35	83.54	73.96$_{\pm 0.36}$	95.95	96.30	96.26	96.17$_{\pm 0.14}$
		IBKE	97.12	**96.23**	97.41	96.92$_{\pm 0.19}$	91.87	47.85	69.86$_{\pm 0.20}$	91.66	55.03	77.47	74.72$_{\pm 0.25}$	93.51	94.68	96.75	94.98$_{\pm 0.28}$
		SERAC*	99.54	83.82	85.65	89.67$_{\pm 0.22}$	99.72	46.79	73.26$_{\pm 0.05}$	98.88	76.00	88.83	87.90$_{\pm 0.24}$	99.08	97.47	99.67	98.74$_{\pm 0.22}$
		MEND*	95.08	72.18	95.27	87.51$_{\pm 0.32}$	93.83	73.24	83.53$_{\pm 0.25}$	94.90	65.16	89.59	83.22$_{\pm 0.11}$	95.85	96.86	97.98	96.90$_{\pm 0.22}$
Qwen3 (8B)	ET	IBKE*	97.48	96.20	97.43	**97.03**$_{\pm 0.19}$	96.00	**88.75**	**92.38**$_{\pm 0.20}$	96.34	**84.26**	93.77	**91.46**$_{\pm 0.22}$	97.64	97.16	98.36	97.72$_{\pm 0.15}$

"W/O" indicates results for pre-edit LLMs. "Rel.", "Gen.", and "Loc." denote reliability, generality, and locality, respectively. **DE** and **ET** refer to editor types based on Direct Editing and Edit Training, respectively. The training dataset for ET methods marked with an asterisk (*) is augmented with a small amount of additional data. The best performance is highlighted in bold, and results within 3% of the best are underlined.

coefficient β, as shown in Fig. 3. Increasing l_m generally leads to improved editing performance, as it directly expands the capacity of the latent representation. The coefficient β governs the generalization ability of the latent representation by mediating the trade-off between information compression and retention.

Unlike the effect of increasing l_m, editing performance does not improve monotonically with larger β values. For $l_m > 1$, reliability and locality typically reach their maximum at $\beta = 0.1$, while generality is greatest at $\beta = 1$. However, setting β to excessively large values (e.g., $\beta = 10$) diminishes all three metrics, as the latent representations become too similar and the flow of useful information is overly restricted. Conversely, when $l_m = 1$, increasing β monotonically degrades editing performance, likely due to insufficient latent representation capacity for effective compression.

Based on these observations, our selection strategy is as follows: employ a larger l_m and choose $\beta = 0.1$ or $\beta = 1$, depending on the desired degree of generalization. Higher values of β enhance generality but may reduce the model's fidelity to the original edit (as measured by the top-1 hit rate). To balance performance and computational efficiency, we select $l_m = 10$ and $\beta = 0.1$ for the remaining experiments in this study.

Ablation Study

As shown in Fig. 4, we first analyze the impact of the two key components of IBKE: the IB mechanism and the scale factor $f_{W_s}(\widetilde{s}_i)$.

For reliability, removing either component leads to only minor decreases in performance. This outcome occurs because the input gradient signal is sufficiently informative to guide the editor in aligning the model with the edit request, even without deeper semantic processing or additional contextual information. This result is consistent with findings from MEND[17], which also achieves high reliability through the use of multilayer perceptrons (MLPs) to independently adjust token-level gradient decompositions. The IB mechanism improves IBKE's capacity to interpret edit semantics and extract key information by compressing the edit signal. When omitted, redundant information remains within the latent representation, impairing generalization and making it more challenging for the scale factor to focus on the most informative token positions. The scale factor modulates the strength of token-level gradient updates based on the extracted semantic features of the edit. By highlighting important token positions and suppressing less pertinent ones, it promotes more targeted adjustments. Its absence causes all token updates to be applied with uniform strength, resulting in diminished generality and locality. In summary, the synergy between the IB mechanism and the scale factor enables the editor to isolate crucial edit information while filtering out unproductive updates, thereby enhancing both generality and locality.

As shown in Table 4, we further conduct an ablation study on the number of augmented samples. As the number of augmented samples increases from 0 to 512, editing performance improves across the three datasets, with the most notable gains in generality and locality observed on MQuAKE and CounterFact. Meanwhile, reliability consistently remains at near-saturated levels. When the number of augmented samples is further increased to 1,024 and 2,048, the improvements exhibit diminishing returns. Overall, approximately 512–1,024 additional augmented samples for each dataset are sufficient to largely unlock the editing potential of IBKE.

Granular Performance Evaluation of the IB Mechanism

To further investigate the performance improvement provided by the IB mechanism in different scenarios, we conducted a more granular analysis across multiple domains and evaluation criteria, as shown in Fig. 5. Each editor instance was trained on a knowledge domain containing fewer than 9,000 samples on average, resulting in overall performance slightly lower than that of the main experiment.

Subfigure 5a illustrates the performance of IBKE with and without IB across various knowledge domains. Compared to locality, the performance difference is more significant in the areas of reliability and generality, where editors with IB applied generally exhibit higher background brightness, corresponding to higher scores. Notably, in more challenging cross-domain

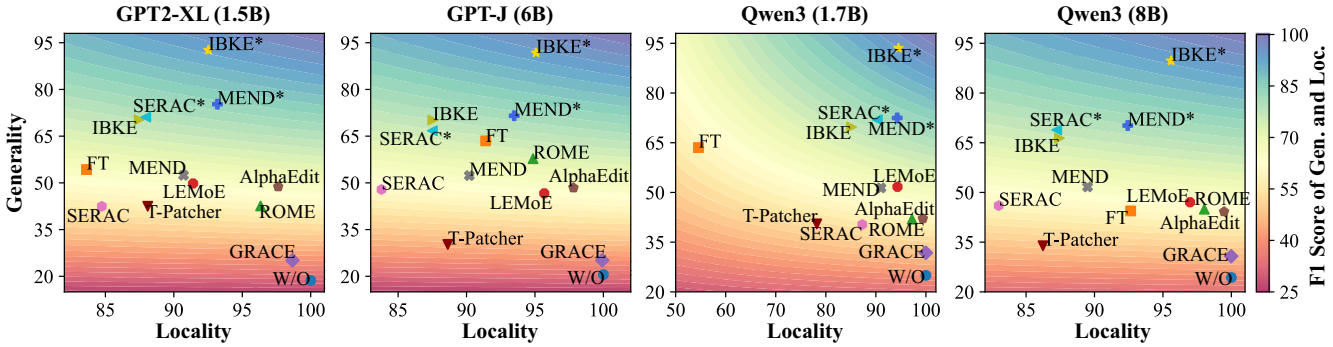


Fig. 2 | The trade-off between generality and locality is illustrated for various editing methods and model backbones. Each data point represents the average generality and locality scores achieved by a given editor, computed as the mean over the UniEdit, MQuAKE, and CounterFact datasets.

Table 3 | Mass editing performance after 3000 edits on a mixed dataset evenly sampled from UniEdit, MQuAKE, CounterFact, and ZSRE

Backbones	Type	Editors	Rel.	Gen.	Loc.	F1	Average
		W/O	21.57	22.41	**100.00**	36.61	$47.99_{\pm 0.22}$
	Base	**FT**	45.87	34.78	29.21	31.76	$36.62_{\pm 1.00}$
		ROME	0.62	0.56	1.85	0.86	$1.01_{\pm 0.25}$
		T-Patcher	60.16	42.33	45.17	43.70	$49.22_{\pm 0.46}$
		LEMoE	21.96	22.64	90.79	36.25	$45.13_{\pm 0.37}$
		GRACE	97.17	30.83	98.97	47.01	$75.66_{\pm 0.25}$
		MEMIT	65.89	38.44	89.36	53.76	$64.56_{\pm 0.52}$
	DE	**AlphaEdit**	82.69	43.51	78.20	55.91	$68.13_{\pm 1.78}$
		SERAC	86.65	49.41	77.27	60.27	$71.11_{\pm 0.17}$
		MEND	80.99	58.35	81.78	68.10	$73.71_{\pm 0.20}$
		IBKE	82.52	66.73	81.26	73.28	$76.84_{\pm 0.15}$
		SERAC*	**98.59**	68.55	78.80	73.32	$81.98_{\pm 0.30}$
		MEND*	82.88	70.60	84.44	76.90	$79.31_{\pm 0.48}$
GPT2-XL (1.5B)	**ET**	**IBKE***	84.12	**82.40**	87.76	**84.99**	$\mathbf{84.76}_{\pm 0.31}$
		W/O	27.10	28.97	**100.00**	44.92	$52.02_{\pm 0.12}$
	Base	**FT**	72.10	51.43	44.04	47.45	$55.85_{\pm 1.67}$
		ROME	48.26	27.27	43.74	33.59	$39.76_{\pm 2.73}$
		T-Patcher	39.72	26.93	12.95	17.49	$26.53_{\pm 0.42}$
		LEMoE	25.67	30.30	93.96	45.82	$49.97_{\pm 1.45}$
		GRACE	**98.25**	34.25	99.96	51.02	$77.49_{\pm 0.18}$
		MEMIT	70.79	41.96	95.84	58.36	$69.53_{\pm 0.48}$
	DE	**AlphaEdit**	71.20	40.07	86.96	54.86	$66.08_{\pm 0.41}$
		SERAC	93.95	52.79	71.04	60.57	$72.59_{\pm 0.13}$
		MEND	80.04	55.66	84.36	67.07	$73.35_{\pm 0.10}$
		IBKE	79.52	65.23	80.67	72.13	$75.14_{\pm 0.36}$
		SERAC*	98.23	68.93	76.33	72.44	$81.16_{\pm 0.24}$
		MEND*	81.13	70.63	90.58	79.37	$80.78_{\pm 0.14}$
Qwen3 (8B)	**ET**	**IBKE***	82.54	**82.47**	93.05	**87.44**	$\mathbf{86.02}_{\pm 0.23}$

Results are averaged over three runs of data sampling, editing, and evaluation.

evaluations, the introduction of IB tends to yield a greater enhancement effect. For example, for both editors trained on the Literature domain, when evaluated on the Biology domain, the reliability difference reaches 4.5, while the generality difference exceeds 3.5 when tested across nearly all domains. The left part of Fig. 5b intuitively illustrates this enhancement. The right part shows that IB effectively shifts the performance distribution toward higher values, making it more concentrated near the upper bound of the original performance. These findings suggest that IB can effectively facilitate the editor's cross-domain editing performance.

Subfigure 5c presents the enhancement effect of IB on the editor from the perspective of generality evaluation criteria. A clear stair-step pattern is observed, with scores decreasing as each additional criterion, such as Multi-Hop and Relation Reverse, is included. Similar to Subfigure 5a, IB provides a stronger enhancement of the editor's generality under more

difficult criteria, and a similar effect is observed as the number of hops increases.

Overall, simpler training domains and more difficult test cases result in lower editing performance, while IB effectively mitigates both challenges. We attribute the gains to two aspects, respectively: IB promotes the learning of more generalized information extraction patterns, thereby improving cross-domain editing performance; the more compact editing representation leaves greater semantic understanding and transformation capacity for the editor, enabling more effective learning of challenging editing generalization.

Feature Visualization and Instance Analysis

To illustrate the semantic representation learning capabilities of our editor architecture, we visualize the distribution of its latent representations extracted by the learnable fixed-length sequence using dimensionality reduction via t-SNE[45], as shown in Fig. 6. We observe that, regardless of whether the IB mechanism is employed, edit samples sharing the same relation tend to cluster together, indicating that both editor variants effectively capture the distribution of edit semantics through the learned representations. In layers 15 and 16, these clusters are well separated, whereas in layer 17, some clusters exhibit less distinct boundaries, particularly for the relations "has part(s)", "used by", and "facet of". We hypothesize that this results from differences in the semantic level encoded by the gradient decomposition of GPT2-XL within this layer. Moreover, incorporating the IB mechanism produces clearer boundaries between clusters and fewer outliers, suggesting improved learning of distinct and well-separated latent representations.

Figure 7 further visualizes the scale factors $f_{W_s}(\tilde{s}_i)$ to assess whether IBKE attends to key positions in the input edit signals, and to examine how the addition of the IB mechanism affects this process. Subfigure 7a shows the distribution of scale factors for edit samples from three different datasets, processed by editor instances with and without the IB mechanism. At the token level, we find that the first token, subject, predicate, and prepositions within an edit sentence receive varying attention across layers. Scale factors in layers 15 and 16 exhibit greater sparsity, while those in layer 17 are more densely distributed. This pattern indicates that these key positions convey most of the semantics necessary for effective editing. Comparing editor instances, we find that scale factors in models trained with the IB mechanism are noticeably sparser, suggesting that IB-driven compact representations enhance the sparsity and selectivity of gradient decompositions.

Figure 7b depicts how editing performance changes as scale factors are systematically pruned from the smallest to the largest values, providing an intuitive demonstration of the advantages of increased sparsity. Editors utilizing IB show more moderated confidence in both reliability and generality, while those lacking IB may over-edit or affect unrelated samples, as reflected by reduced locality similarity. Additionally, as shown in the "Average" column, editors with IB exhibit slower increases in prediction rank for generality samples during early pruning, whereas editors without IB experience a pronounced drop in locality similarity during later stages. Overall, these results suggest that the IB mechanism mitigates interference from redundant, entangled token positions during editing, contributing to greater robustness and superior overall performance.

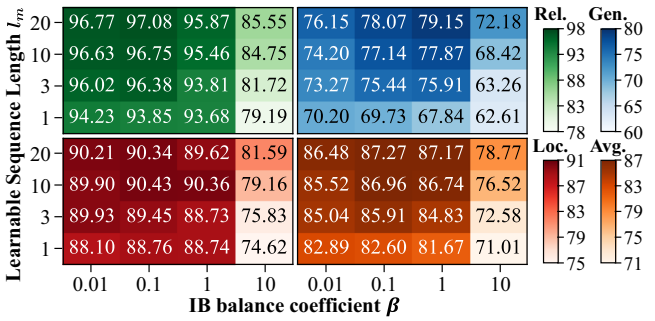


Fig. 3 | Hyperparameter search for the learnable sequence length l_m and the IB trade-off coefficient β, using GPT2-XL as the backbone. IBKEs with different configurations are trained on UniEdit, and the results show the average performance across the four datasets.

Computational Overheads and Training Difficulty

Figure 8 compares the computational overhead of different editors across LLMs of varying scales from the perspective of single-edit cost. In terms of editing time, DE-based methods are relatively expensive, as each edit

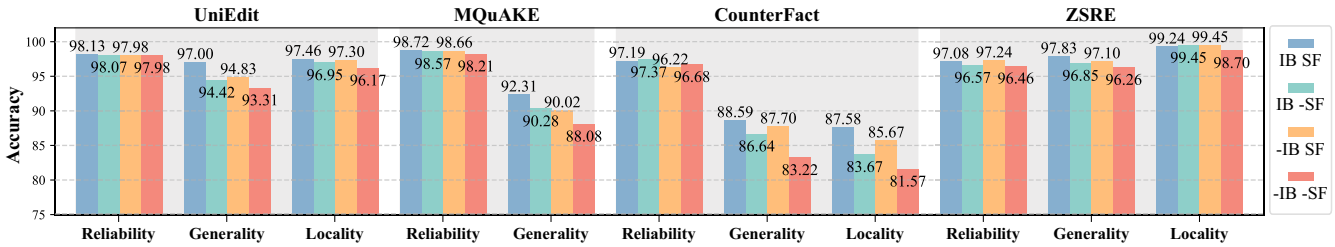
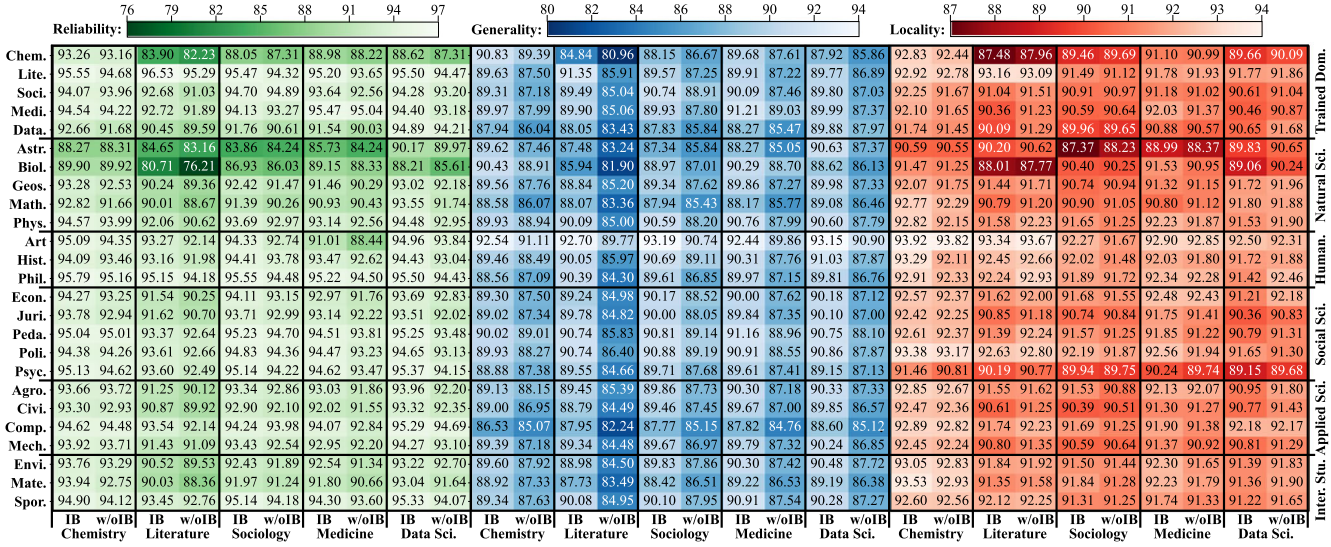


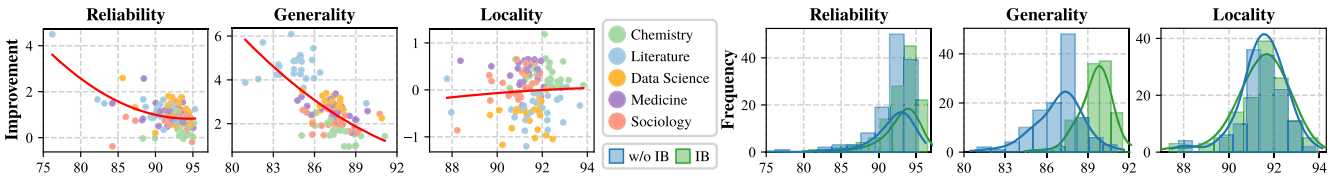
Fig. 4 | Ablation study of the IB mechanism and the scale factor (SF) $f_{W_s}(\tilde{s}_i)$, using GPT2-XL as the backbone. The four IBKE variants with different configurations are trained on the few-shot augmented data. Legends marked with a "-" sign indicate that the corresponding module has been removed.

Table 4 | IBKE ablation on data augmentation amounts, where 0 and 512 augmented samples correspond to IBKE and IBKE*, respectively

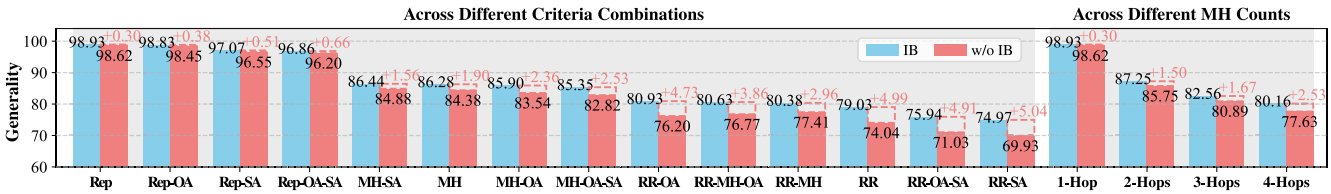
	UniEdit				MQuAKE			CounterFact				ZSRE			
# Data	Rel.	Gen.	Loc.	Average	Rel.	Gen.	Average	Rel.	Gen.	Loc.	Average	Rel.	Gen.	Loc.	Average
0	97.80	96.89	97.48	97.39$_{\pm 0.30}$	97.34	56.09	76.72$_{\pm 0.25}$	96.08	57.87	77.45	77.13$_{\pm 0.18}$	95.79	97.69	96.36	96.61$_{\pm 0.32}$
128	98.12	96.29	97.26	97.22$_{\pm 0.26}$	98.09	74.69	86.39$_{\pm 0.18}$	96.10	63.47	85.47	81.68$_{\pm 0.08}$	95.43	97.78	97.67	96.96$_{\pm 0.13}$
256	97.92	96.26	97.47	97.22$_{\pm 0.23}$	98.65	86.36	92.51$_{\pm 0.39}$	96.81	69.83	84.97	83.87$_{\pm 0.48}$	96.05	98.13	98.29	97.49$_{\pm 0.28}$
512	**98.13**	97.00	97.46	97.53$_{\pm 0.17}$	98.72	92.31	95.51$_{\pm 0.43}$	97.19	88.59	87.58	91.12$_{\pm 0.22}$	97.08	97.83	99.24	98.05$_{\pm 0.13}$
1024	98.11	**97.06**	**97.56**	**97.58**$_{\pm 0.24}$	99.09	92.98	96.03$_{\pm 0.61}$	96.87	93.20	94.33	94.80$_{\pm 0.15}$	97.70	99.13	99.69	98.84$_{\pm 0.17}$
2048	98.06	96.07	97.53	97.22$_{\pm 0.14}$	**99.50**	**93.72**	**96.61**$_{\pm 0.47}$	**97.77**	**96.40**	**96.10**	**96.76**$_{\pm 0.20}$	**98.32**	**99.23**	**99.78**	**99.11**$_{\pm 0.06}$



(a) Overall performance across the 25 domains.



(b) Performance gains from IB across domains.



(c) Average generality across different criteria.

Fig. 5 | Performance of IBKE with and without the IB mechanism, trained on five domains from different sectors in UniEdit, using Qwen3-1.7B as the backbone. **a** Shows the overall performance of each training instance across the 25 domains in UniEdit, where the vertical axis represents the 25 test domains, and the horizontal axis represents the training domains. **b** Visualizes the performance gains from IB across domains in the left part, where different colors represent different training domains. Each point represents a test result on a specific domain, with the x-axis showing the performance before introducing IB. A quadratic curve is fitted to the data to highlight the trend. The right part displays the distribution of performance with and without IB. **c** Shows the average generality over the five training instances across different combinations of criteria and hop counts in UniEdit, where the abbreviations denote Rephrase (Rep), Object Alias (OA), Subject Alias (SA), Multi-Hop (MH), and Relation Reverse (RR).

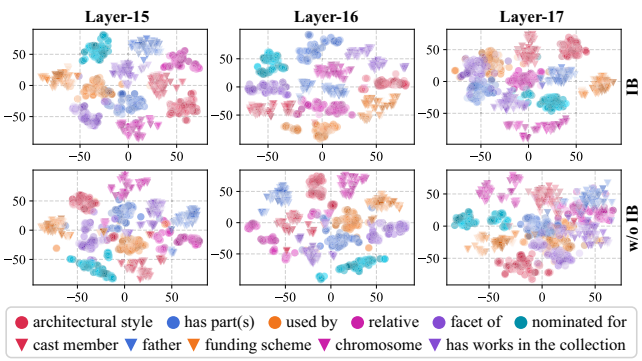
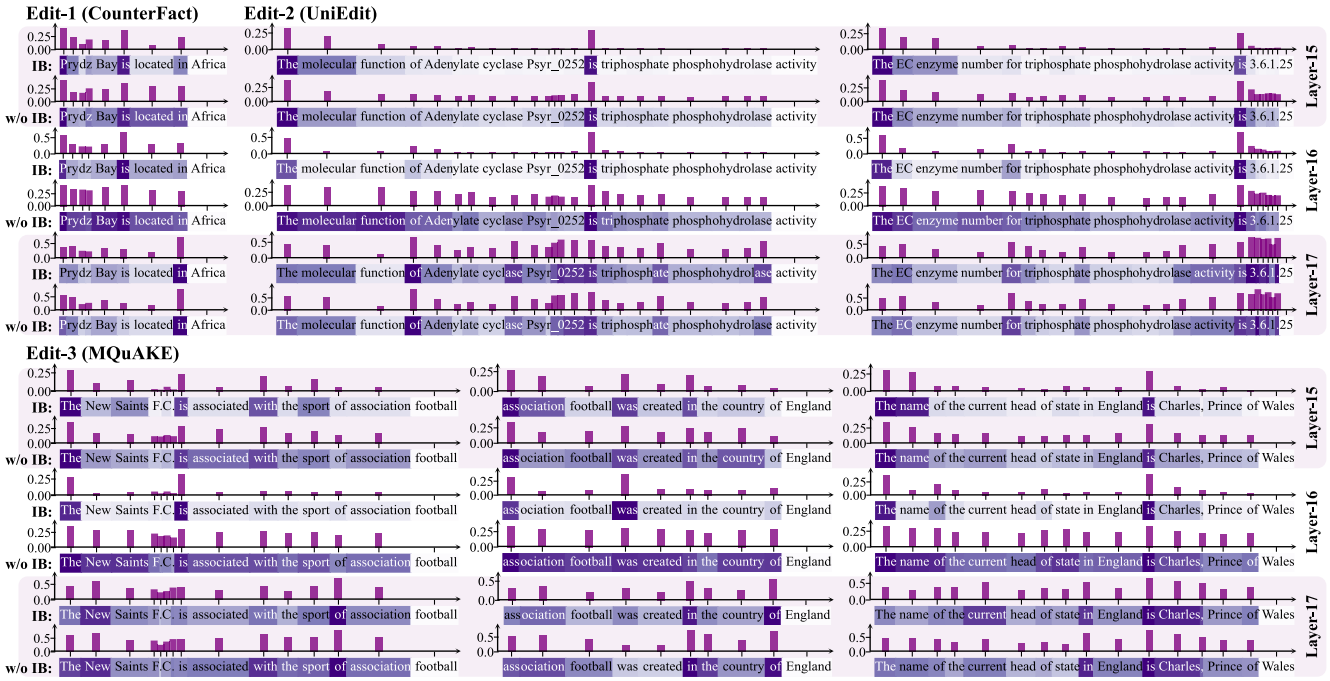


Fig. 6 | Latent representation visualization of IBKE with and without the IB mechanism using t-SNE, with GPT2-XL as the backbone. Each point represents an edit sample, and each class corresponds to samples sharing the same relation. Brightness indicates the prediction confidence of the edited model given the edited samples.

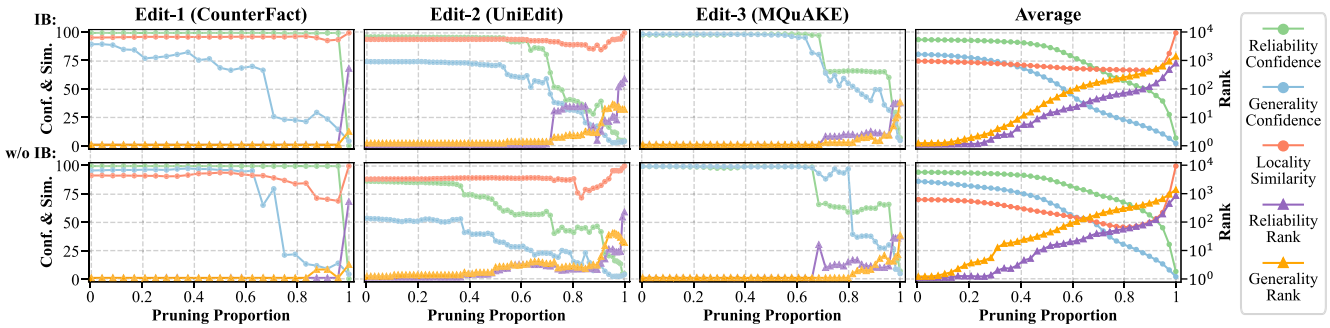
requires multiple iterations of training, resulting in latency on the order of seconds. Regarding memory consumption, ROME and AlphaEdit incur relatively high GPU memory overhead, as they must backpropagate gradients to intermediate layers of the LLM while maintaining high-dimensional statistical matrices and weight snapshots. In contrast, ET-based methods such as MEND, SERAC, and IBKE achieve at least an order-of-magnitude reduction in editing latency by avoiding repeated back-propagation. Moreover, their memory overhead remains well-controlled across backbone models of different scales, indicating better scalability. Notably, SERAC directly stores edit statements during the editing process, while the counterfactual small language model used for response generation remains fixed. This design results in low editing latency and slow growth in memory consumption.

Figure 9 illustrates the evolution of loss and performance during the training of IBKE. As training progresses, both training and validation losses decrease steadily and eventually converge, indicating that the objective function with the IB constraint is well-optimized for the knowledge editing task and does not exhibit noticeable instability or divergence. Meanwhile, all three evaluation metrics improve during the early to middle stages of



(a) Scale factor values across layers for each edit token.

- **Edit-1 Generality Sample (1-Hop):** The Bulletin … evening newscasts. Prydz Bay belongs to the continent of Africa
- **Edit-1 Locality Sample :** Antarctic Treaty System belongs to the continent of Antarctica
- **Edit-2 Generality Sample (2-Hops):** The EC enzyme number for the molecular function of Adenylate cyclase Psyr_0252 is 3.6.1.25
- **Edit-2 Locality Sample:** Is the activity that has hydrogen phosphate ion as part(s) the same as the activity that has water as part(s)? Yes
- **Edit-3 Generality Sample (3-Hops):** Who is the head of state of the country where The New Saints F.C. was founded? Charles, Prince of Wales



(b) Changes in edit metrics as scale factors are pruned.

Fig. 7 | Visualization of edit token importance for IBKE with and without the IB mechanism, using GPT-2-XL as the backbone. The three edit examples are drawn from CounterFact, UniEdit, and MQuAKE, respectively. a Shows the strength of scale factors $f_{W_v}(\tilde{s}_i)$ across layers for each edit token (values are activated by the sigmoid function), where darker token backgrounds intuitively correspond to higher values. **b** Displays changes in edit metrics as scale factors are iteratively pruned to zero, from the lowest to the highest values. Both the average prediction confidence and the vocabulary rank of the target tokens are reported for reliability and generality. Locality similarity is computed based on the JS divergence between the edited and original models' prediction distributions on the locality samples. "Average" reports the averaged results over 3000 successfully edited instances randomly drawn from the three datasets.

training, followed by a plateau phase, with almost no performance degradation observed in the later stages. This phenomenon indicates that IBKE can effectively learn reliable fitting of editing targets while maintaining cross-sample generalization, thereby alleviating or delaying the onset of potential overfitting.

Discussion

Current editors for LLMs are often effective within narrowly defined or constrained settings, but they frequently struggle to generalize in open, real-world scenarios[21,22,25,37,38]. In this work, we introduce a theoretically principled framework for model editing that explicitly manages the balance between generality and locality. By framing model editing as an information-constrained optimization problem, this approach distinguishes essential edit information from redundant signals, enabling effective generalization of edits while reducing interference with unrelated knowledge.

Building on this foundation, we develop the Information Bottleneck Knowledge Editor (IBKE), which implements the information bottleneck (IB) principle through gradient-based latent encoding and adaptive token-level scaling. Empirical results drawn from four datasets, four backbone architectures, and nine baseline methods, comprising 208 evaluation experiments in total, demonstrate that IBKE achieves leading performance in balancing generality and locality.

Extensive hyperparameter exploration reveals that a moderate balance coefficient ($\beta \approx 0.1$) strikes an optimal compromise between reliability and generality, providing new insight into how the IB principle guides model editing. Ablation studies demonstrate that both the IB mechanism and the token-level scale factor are critical; removal of either component results in redundant or misaligned updates. This underscores that information compression and adaptive scaling are key for IBKE to focus on semantically relevant changes and to precisely execute edits.

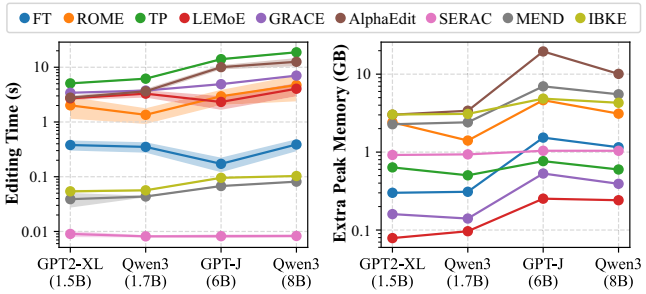


Fig. 8 | Editing time and extra peak memory overhead of different editors across four LLM backbones. Experiments are conducted on a single A6000 GPU and one CPU core. Both model and editor parameters use single precision (FP32).

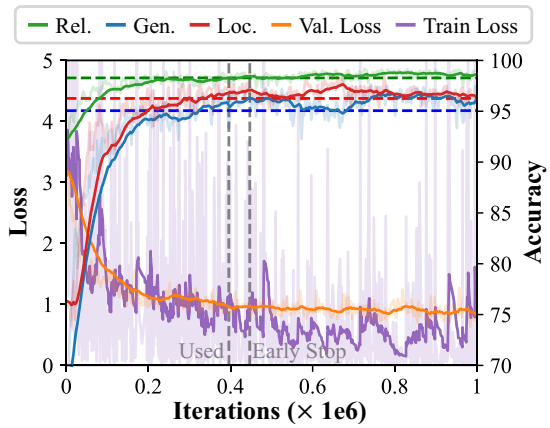


Fig. 9 | Training and validation loss, along with the average reliability, generality, and locality on the four test datasets, during continued training of IBKE* on Qwen3-1.7B up to 1M steps. The two vertical dashed lines mark the checkpoint used for main experiments and the early stopping step, while horizontal dashed lines of different colors indicate the average test results of each metric in the main experiments.

Granular evaluations across knowledge domains and evaluation criteria show that editors incorporating the IB mechanism demonstrate enhanced cross-domain generalization and greater robustness under challenging conditions, indicating that IB supports the learning of transferable editing patterns. Visualization analyses further reveal that IB increases the separability of latent semantic clusters and promotes sparsity in token-level updates, confirming that the editor effectively targets and utilizes the most informative aspects of the edit. These findings support our central hypothesis: constraining information flow enables more robust, balanced, and interpretable model updating. Additionally, we compare computational overhead across editors and analyze the training dynamics of IBKE, demonstrating its moderate cost, efficient convergence, and robustness against overfitting.

Overall, our study demonstrates that model editing within the IB framework offers a promising approach for achieving reliable, generalizable, and interpretable updates in LLMs, with IBKE representing a significant advance in this direction. One limitation of our current implementation is the relatively high-rank nature of weight updates. Future work may seek to integrate the IB principle with low-rank adaptation strategies (e.g., rank-1 updates) to further reduce potential interference with original model parameters. In addition, the current extension of IBKE to mass editing relies on hard retrieval, where retrieval accuracy may weaken interactions among related edits. Addressing the integration of mass edits within the IB-based editing framework therefore represents a promising direction for enabling more effective lifelong editing.

Methods

In this section, we first describe the preliminaries of model editing and the IB principle. We then introduce the IB-enhanced model editing framework and its information flow, followed by a detailed description of IBKE, which is constructed based on this framework.

Preliminaries

For model editing, we represent an LLM as $f_\theta \in \mathcal{F}$, where f_θ: $Q \to O$ maps an input query $q \in Q$ to an output $o = f_\theta(q)$. An edit request $e = (q_e, o_e)$ instructs the LLM to produce the target output o_e when given the prompt q_e, particularly when $f_\theta(q_e) \neq o_e$. Given a set of edit requests $\mathcal{E} = \{e_i\}_{i=1}^n$, model editing seeks to construct an editor $\phi : \mathcal{F} \times \mathcal{E} \to \mathcal{F}$ that generates a post-edit LLM, denoted $f_{\hat{\theta}_\mathcal{E}} = \phi(f_\theta, \mathcal{E})$. An effective editor should ensure that the edited LLM satisfies the following three metrics[14]:

Reliability assesses the accuracy of the responses generated by $f_{\hat{\theta}_\mathcal{E}}$ for the edited samples:

$$\mathbb{E}_{(q_e,o_e)\in\mathcal{E}}[\mathbb{I}(f_{\hat{\theta}_\mathcal{E}}(q_e) = o_e)] \tag{1}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

Generality measures whether $f_{\hat{\theta}_\mathcal{E}}$ appropriately adapts its responses to queries related to the edited samples:

$$\mathbb{E}_{(q_g,o_g)\sim\mathcal{G}(\mathcal{E})}[\mathbb{I}(f_{\hat{\theta}_\mathcal{E}}(q_g) = o_g)] \tag{2}$$

where $\mathcal{G}(\mathcal{E})$ denotes the relevant neighborhood (the "ripple effect") of the edit set $\mathcal{E}$.

Locality evaluates whether $f_{\hat{\theta}_\mathcal{E}}$ maintains responses consistent with the original model for queries unrelated to the edited set:

$$\mathbb{E}_{(q_l,o_l)\sim\mathcal{L}(\mathcal{E})}[\mathbb{I}(f_{\hat{\theta}_\mathcal{E}}(q_l) = f_\theta(q_l))] \tag{3}$$

where $\mathcal{L}(\mathcal{E})$ refers to the set of queries unrelated to $\mathcal{E}$, specifically excluding $\mathcal{E} \cup \mathcal{G}(\mathcal{E})$.

For IB, we consider a supervised learning setting with random variables (X, Y), where X denotes the model input and Y represents the corresponding ground-truth label. The IB principle[33,34] models the forward propagation process ϕ as a Markov chain:

$$Y \to \underbrace{X \to Z \to \hat{Y}}_{\phi} \tag{4}$$

Here, Z is an intermediate representation of X produced by the model, and $\hat{Y}$ is the predicted label. This structure implies that Y and Z are conditionally independent given X, indicating that all information relevant to Y must pass through X.

The IB objective seeks to learn a representation Z that is both compressed and sufficient, by minimizing the mutual information $I(Z; X)$ while maximizing $I(Z; Y)$:

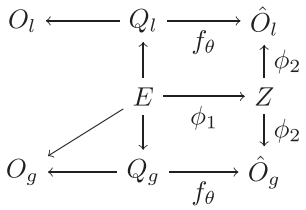
$$\max I(Z; Y) - \beta I(Z; X) \tag{5}$$

where $\beta > 0$ is a trade-off parameter that determines the level of compression. By balancing these two terms, the IB principle encourages the model to extract essential information from X for predicting Y, while reducing redundant content.

Knowledge Editing: an IB Perspective

In this subsection, we present a unified formulation of model editing as a Markov chain, which provides the foundation for the IB approach and its

associated optimization objectives. Unlike conventional supervised learning tasks, the forward pass here consists of two distinct stages: first, intervention by the model editor in the LLM, and second, generation of responses by the edited LLM to subsequent inputs. We describe the information flow among relevant variables as follows:



(6)

Within this framework, each edit induces specific distributions for generality and locality, meaning that both generality samples (Q_g, O_g) and locality samples (Q_l, O_l) are conditioned on the edit request E. We explicitly partition the editor ϕ into two stages: ϕ_1, which derives the latent representation Z from the edit request E, and ϕ_2, which applies Z to intervene in the LLM, thereby altering its predicted response $\widehat{O}$ to the input Q. Based on these dependencies, we propose three constraints for the model editing process to achieve effective editing while minimizing the transmission of redundant information.

The first constraint is Information Transfer Minimization (ITM), which requires the editor to compress the information contained in the representation Z as much as possible. Formally, this is expressed as

$$\min I(E; Z). \tag{7}$$

Expanding the mutual information yields:

$$\sum_{e,z} p(e,z) \log \frac{p(z|e)}{p(z)}. \tag{8}$$

Since $p(z)$ is generally intractable, by introducing a reference distribution $r(z)$ and leveraging the non-negativity of the KL divergence, we obtain the following variational upper bound:

$$\sum_{e,z} p(e,z) \log \frac{p(z|e)}{p(z)} \le \sum_{e,z} p(e,z) \log \frac{p(z|e)}{r(z)}. \tag{9}$$

In the context of Eq. (6), the distribution $p(z|e)$ is parameterized as $p_{\phi_1}(z|e)$. Thus, the objective becomes optimizing the upper bound with respect to ϕ_1:

$$\min_{\phi_1} \sum_{e,z} p(e) p_{\phi_1}(z|e) \log \frac{p_{\phi_1}(z|e)}{r(z)} \tag{10}$$

$$\Rightarrow \min_{\phi_1} \sum_{e} p(e)\, KL[p_{\phi_1}(z|e) \parallel r(z)] \tag{11}$$

The second constraint is Sufficiency for Generality (SG), which requires the representation Z and the generality input Q_g to retain as much information as possible about the generality output O_g. This can be formulated and derived as follows:

$$\max I(Z, Q_g; O_g) \tag{12}$$

$$\Rightarrow \max \sum_{z,q_g,o_g} p(z,q_g,o_g) \log \frac{p(o_g|z,q_g)}{p(o_g)} \tag{13}$$

$$\Rightarrow \max \sum_{z,q_g,o_g} p(z,q_g,o_g) \log p(o_g|z,q_g) + H(O_g) \tag{14}$$

$$\propto \max \sum_{z,q_g,o_g} p(z,q_g,o_g) \log p(o_g|z,q_g) \tag{15}$$

Since the entropy $H(O_g)$ is constant, it is dropped in the optimization. By replacing $p(o_g|z,q_g)$ with an approximating distribution $\xi(o_g|z,q_g)$ and using the non-negativity of the KL divergence, we obtain the following variational lower bound:

$$\sum_{z,q_g,o_g} p(z,q_g,o_g) \log p(o_g|z,q_g) \ge \sum_{z,q_g,o_g} p(z,q_g,o_g) \log \xi(o_g|z,q_g) \tag{16}$$

In the context of Eq. (6), $\xi(o_g|z,q_g)$ is modeled by the decoder formed as the composition of ϕ_2 and f_θ. Thus, the objective becomes optimizing the lower bound with respect to ϕ, as derived below:

$$\max_{\phi} \sum_{z,q_g,o_g} p(z,q_g,o_g) \log \xi(o_g|z,q_g) \tag{17}$$

$$\Rightarrow \max_{\phi} \sum_{z,e,q_g,o_g} p(z,e,q_g,o_g) \log \xi(o_g|z,q_g) \tag{18}$$

$$\Rightarrow \max_{\phi} \sum_{z,e,q_g,o_g} p(e)\, p(q_g,o_g|e)\, p_{\phi_1}(z|e) \log \xi(o_g|z,q_g) \tag{19}$$

The derivation of Eq. (19) follows from the conditional independence of q_g, o_g and z given e, as shown in Eq. (6).

The third constraint is Independence for Locality (IL), which requires the representation Z, given the locality input Q_l, to contain as little information as possible about the locality output $\widehat{O}_l$ of the LLM. This is formulated as follows:

$$\min I(Z; \hat{o}_l|Q_l) \tag{20}$$

$$\Rightarrow \min \sum\nolimits_{z,\hat{o}_l,q_l} p(z,\hat{o}_l,q_l) \log \frac{p(\hat{o}_l|z,q_l)}{p(\hat{o}_l|q_l)} \tag{21}$$

In the context of Eq. (6), the distribution $p(\hat{o}_l|z,q_l)$ is parameterized as $\xi(\widehat{o}_l|z,q_l)$.

For the conditional distribution $p(\widehat{o}_l|q_l)$, we replace it with the original model f_θ, which is unaffected by Z, and similar to Eq. (9), we obtain a variational upper bound. The objective then becomes:

$$\min_{\phi} \sum_{z,\widehat{o}_l,q_l} p(z,\hat{o}_l,q_l) \log \frac{\xi(\hat{o}_l|z,q_l)}{f_\theta(\hat{o}_l|q_l)} \tag{22}$$

$$\Rightarrow \min_{\phi} \sum_{e,z,\widehat{o}_l,q_l} p(z,e)\, p(\hat{o}_l|q_l,z,e)\, p(q_l|z,e) \cdot \log \frac{\xi(\hat{o}_l|z,q_l)}{f_\theta(\hat{o}_l|q_l)} \tag{23}$$

$$\Rightarrow \min_{\phi} \sum_{e,z,\widehat{o}_l,q_l} p(e)\, p(q_l|e)\, \xi(\hat{o}_l|q_l,z)\, p_{\phi_1}(z|e) \cdot \log \frac{\xi(\hat{o}_l|z,q_l)}{f_\theta(\hat{o}_l|q_l)} \tag{24}$$

In Eq. (24), the terms $p(q_l|e)$ and $\xi(\widehat{o}_l|q_l,z)$ arise from conditional independence assumptions, which allow certain variables to be omitted.

Information Bottleneck Knowledge Editor (IBKE)

In this subsection, we introduce IBKE, an editor built on the IB framework described above. Inspired by previous studies on LLM knowledge localization and editing[19,21,22,24], IBKE corrects the responses of an LLM by

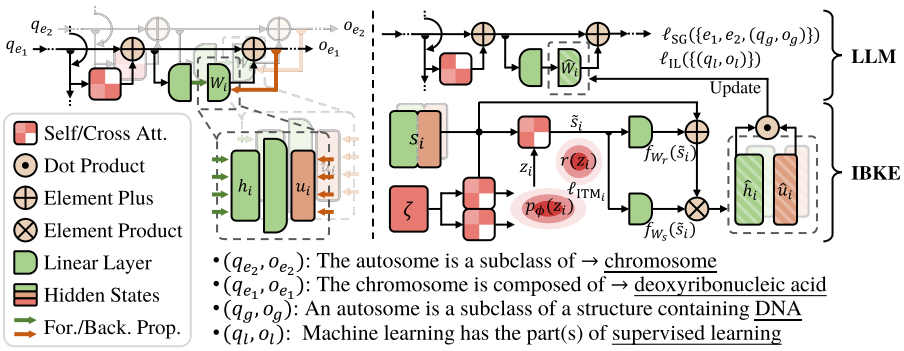


Fig. 10 | Overall structure of IBKE. A batch of two-hop edit samples is shown as an example. The left panel illustrates how edit signals are obtained, while the right panel depicts the pipeline by which IBKE utilizes these signals to update model weights.

adapting the weights of the output linear layers within the early-layer FFNs. To achieve this, a set of hypernetworks is trained; each hypernetwork receives as input a signal that encodes the prior of the edit request and is used to generate weight offsets. The full computational workflow of IBKE is illustrated in Fig. 10.

The first stage of IBKE, denoted as ϕ_1, computes the first-order gradient decompositions of the edit weights with respect to the edit request[17,32,46], which are then used as inputs to the associated hypernetworks. While simpler alternatives are available, such as using embeddings or hidden representations of the edit request[20,29], the gradient decomposition conveys not only the semantics of the edit but also the direction of steepest weight adaptation.

Specifically, within the LLM f_θ, we consider a set of linear layer matrices $\{W_i \in \mathbb{R}^{d_{in} \times d_{out}}\}_{i=1}^{n}$, each paired with a hypernetwork that processes its corresponding gradient signal. Here, d_{in} and d_{out} are the input and output dimensions of each linear layer, respectively. Given an edit request $e = (q_e, o_e)$, the gradient for each matrix can be decomposed as:

$$\nabla W_i = \frac{\partial \ell_s(q_e, o_e, f_\theta)}{\partial W_i} = h_i^\top u_i = h_i^\top \frac{\partial \ell_s(q_e, o_e, f_\theta)}{\partial (h_i W_i)} \tag{25}$$

where ℓ_s denotes the cross-entropy loss, $h_i \in \mathbb{R}^{l_e \times d_{in}}$ is the input to the linear layer, and $u_i \in \mathbb{R}^{l_e \times d_{out}}$ is the gradient of the linear mapping $h_i W_i$. The sequence length of the edit request is denoted as l_e. Then, the concatenated vector $s_i = h_i \oplus u_i \in \mathbb{R}^{l_e \times (d_{in}+d_{out})}$ serves as the input to the hypernetwork. Below, we describe the computation performed within each hypernetwork.

Given input s_i, IBKE models the latent distribution $p(z_i|s_i)$ and aligns it to a prior distribution $r(z_i)$ using the ITM constraint. To enable differentiable optimization, we employ the reparameterization trick[34,47]. Specifically, $p(z_i|s_i)$ is parameterized as a Gaussian distribution, with two cross-attention (CA) modules mapping s_i to the mean and log-variance of the distribution:

$$\mu_{z_i} = f_{CA_\mu}(\zeta, s_i) \in \mathbb{R}^{l_m \times d_m}, \tag{26}$$

$$v_{z_i} = \sqrt{\exp f_{CA_v}(\zeta, s_i)} \in \mathbb{R}^{l_m \times d_m} \tag{27}$$

where $\zeta \in \mathbb{R}^{l_m \times d_m}$ is a learnable sequence of fixed length l_m, and d_m denotes the module dimension in IBKE.

The cross-attention modules are defined as:

$$f_{CA}(\zeta, s_i) = \delta(\zeta W_q (s_i W_k)^\top) s_i W_v \tag{28}$$

where $W_q \in \mathbb{R}^{d_m \times d_m}$, $W_k \in \mathbb{R}^{(d_{in}+d_{out}) \times d_m}$, and $W_v \in \mathbb{R}^{(d_{in}+d_{out}) \times d_m}$ are projection matrices; biases are omitted for brevity. Here, δ denotes the softmax function.

The prior $r(z_i)$ is chosen as the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Following Eq. (11), the ITM constraint for a request e leads to the following loss, specified by the KL divergence between the two Gaussian distributions:

$$\ell_{\text{ITM}_i}(e) = \frac{1}{2}\{-\mathbf{1}^\top \log \bar{v}_{z_i}^2 + \| \bar{v}_{z_i}^2 \|_1 + \| \bar{\mu}_{z_i}^2 \|_1 - l_m d_m\} \tag{29}$$

where $\bar{v}_{z_i}$ and $\bar{\mu}_{z_i}$ are the flattened versions of v_{z_i} and μ_{z_i}, respectively, and $\mathbf{1}$ is an all-ones vector.

During training, the latent representation $z_i \in \mathbb{R}^{l_m \times d_m}$ is sampled as:

$$z_i = \mu_{z_i} + v_{z_i} \odot \epsilon, \ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{30}$$

where $\epsilon \in \mathbb{R}^{l_m \times d_m}$ is noise sampled from the standard normal distribution, and $\odot$ denotes element-wise multiplication. During inference, we set $\epsilon = 0$ to ensure deterministic outputs.

The second stage of IBKE, denoted as ϕ_2, adapts the LLM f_θ using the gradient decomposition augmented by the latent representation z_i. Specifically, s_i is reparameterized conditional on z_i via a cross-attention module:

$$\tilde{s}_i = f_{CA_s}(s_i, z_i) \in \mathbb{R}^{l_e \times d_m} \tag{31}$$

Next, two separate linear layers are applied to map $\tilde{s}_i$ into: (1) a residual vector $f_{W_r}(\tilde{s}_i) \in \mathbb{R}^{l_e \times (d_{\text{in}}+d_{\text{out}})}$, enhancing s_i; and (2) a scaling factor $f_{W_s}(\tilde{s}_i) \in \mathbb{R}^{l_e \times 1}$, adjusting update strength across tokens. The modified gradient decomposition $\hat{s}_i$ is then computed as:

$$\hat{s}_i = \sigma(f_{W_s}(\tilde{s}_i)) \odot (s_i + f_{W_r}(\tilde{s}_i)) \in \mathbb{R}^{l_e \times (d_{\text{in}}+d_{\text{out}})} \tag{32}$$

where σ denotes the sigmoid function and $\odot$ denotes the element-wise product, applied with broadcasting.

The concatenated vector $\hat{s}_i$ is then split back into $\hat{h}_i \in \mathbb{R}^{l_e \times d_{in}}$ and $\hat{u}_i \in \mathbb{R}^{l_e \times d_{\text{out}}}$. Finally, the edit is executed by updating the matrices $\{W_i\}_{i=1}^{n}$ with the adapted gradients:

$$\hat{W}_i = W_i - \eta_i \hat{h}_i^\top \hat{u}_i, \ i = 1, \ldots, n \tag{33}$$

where η_i is a trainable parameter serving as the learning rate for the edit gradients. For a batch of edit requests, this update is applied iteratively across the batch.

For the training loss of IBKE, given a batch of requests $\mathcal{E}$ and noise ϵ, we denote the IBKE-edited LLM as $f_{\hat{\theta}_\mathcal{E}^\epsilon}$. Using the SG (sufficiency for generality) and IL (independence for locality) constraints and by empirically approximating the true data distribution, we define the batch-wise losses as:

$$\ell_{\text{SG}}(\mathcal{E}) = -\frac{1}{|\mathcal{G}(\mathcal{E})|} \sum_{(q_g, o_g) \in \mathcal{G}(\mathcal{E})} \log f_{\hat{\theta}_\mathcal{E}^\epsilon}(o_g | q_g) \tag{34}$$

$$\ell_{\text{IL}}(\mathcal{E}) = \frac{1}{|\mathcal{L}(\mathcal{E})|} \sum_{(q_l, o_l) \in \mathcal{L}(\mathcal{E})} \text{KL}[f_{\hat{\theta}_\mathcal{E}^\epsilon}(o_l | q_l) \ \| \ f_\theta(o_l | q_l)] \tag{35}$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Here, differing from Eq. (24), $\ell_{IL}(\mathcal{E})$ replaces the autoregressively generated locality output $\widehat{o}_l$ with real outputs o_l from $\mathcal{L}(e)$ to enable parallel token-wise loss computation.

Together with the ITM constraint, the total edit training loss is:

$$\ell_{\text{IBKE}} = \ell_{\text{SG}}(\mathcal{E}) + \ell_{\text{IL}}(\mathcal{E}) + \frac{\beta}{|\mathcal{E}| n l_m d_m} \sum_{e \in \mathcal{E}} \sum_{i=1}^{n} \ell_{\text{ITM}_i}(e) \tag{36}$$

where the denominator $|\mathcal{E}| n l_m d_m$ normalizes the ITM constraint at the element level, preventing the IB regularization term from becoming excessively large.

Data availability

The datasets used in this study are publicly available from the following sources: UniEdit (https://huggingface.co/datasets/qizhou/UniEdit), MQuAKE (https://github.com/princeton-nlp/MQuAKE), CounterFact (https://rome.baulab.info/), ZSRE (https://github.com/eric-mitchell/mend).

Code availability

The code was implemented in Python using the deep learning framework PyTorch (version 2.7.1) and is publicly available at https://github.com/qizhou000/IBKE/.

Received: 9 November 2025; Accepted: 1 May 2026;
Published online: 18 May 2026

References

1. Touvron, H. et al. Llama: Open and efficient foundation language models. *Preprint at arXiv* https://doi.org/10.48550/arXiv.2302.13971 (2023).
2. Roumeliotis, K. I. & Tselikas, N. D. Chatgpt and open-ai models: A preliminary review. *Future Internet* **15**, 192 (2023).
3. Zeng, A. et al. GLM-130B: an open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations* (2023).
4. Luitse, D. & Denkena, W. The great transformer: Examining the role of large language models in the political economy of AI. *Big Data Soc.* **8**, 205395172110477 (2021).
5. Li, N., Gao, C., Li, M., Li, Y. & Liao, Q. Econagent: Large language model-empowered agents for simulating macroeconomic activities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 15523–15536 (2024).
6. Liu, S. et al. Findabench: Benchmarking financial data analysis ability of large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, 710–725 (2025).
7. Nerella, S. et al. Transformers and large language models in healthcare: A review. *Artif. Intell. Med.* **154**, 102900 (2024).
8. Zheng, Y. et al. Large language models for medicine: a survey. *Int. J. Mach. Learn. Cybern.* **16**, 1015–1040 (2025).
9. Wu, C. et al. Towards evaluating and building versatile large language models for medicine. npj Digit. Medicine 8 (2025).
10. Zhou, S. et al. Large language models for disease diagnosis: A scoping review. *npj Artif. Intell.* **1**, 9 (2025).
11. Yan, L. et al. Practical and ethical challenges of large language models in education: A systematic scoping review. *Br. J. Educ. Technol.* **55**, 90–112 (2024).
12. Gao, L. et al. Fine-tuned large language model for visualization system: A study on self-regulated learning in education. *IEEE Trans. Vis. Comput. Graph.* **31**, 514–524 (2025).
13. Raihan, N., Siddiq, M. L., Santos, J. C. S. & Zampieri, M. Large language models in computer science education: A systematic literature review. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education*, 938–944 (2025).
14. Yao, Y. et al. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 10222–10240 (2023).
15. Wang, S. et al. Knowledge editing for large language models: A survey. *ACM Comput. Surv.* **57**, 59:1–59:37 (2025).
16. Zhang, N. et al. A comprehensive study of knowledge editing for large language models. *Preprint at arXiv* https://doi.org/10.48550/arXiv.2401.01286 (2024).
17. Mitchell, E., Lin, C., Bosselut, A., Finn, C. & Manning, C. D. Fast model editing at scale. In The Tenth International Conference on Learning Representations (2022).
18. Huang, J. et al. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 1416–1428 (2024).
19. Dai, D. et al. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 8493–8502 (2022).
20. Chen, Q. et al. Attribution analysis meets model editing: Advancing knowledge correction in vision language models with visedit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2168–2176 (2025).
21. Meng, K., Bau, D., Andonian, A. & Belinkov, Y. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems* **35**, 17359–17372 (2022).
22. Meng, K., Sharma, A. S., Andonian, A. J., Belinkov, Y. & Bau, D. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations* (2023).
23. Li, X. et al. PMET: precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 18564–18572 (2024).
24. Fang, J. *et al*. Alphaedit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations* (2024).
25. Huang, Z. *et al*. Transformer-patcher: One mistake worth one neuron. In *The Eleventh International Conference on Learning Representations* (2023).
26. Zhang, M. *et al*. Uncovering overfitting in large language model editing. In *The Thirteenth International Conference on Learning Representations* (2025).
27. Zhong, Z., Wu, Z., Manning, C. D., Potts, C. & Chen, D. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 15686–15702 (2023).
28. Chen, Q. et al. UniEdit: A unified knowledge editing benchmark for large language models. In *Advances in Neural Information Processing Systems* 38, Datasets and Benchmarks Track https://proceedings.neurips.cc/paper_files/paper/2025/hash/6cc6c2be42c838827d2f1d12baa05539-Abstract-Datasets_and_Benchmarks_Track.html (2025).
29. Cao, N. D., Aziz, W. & Titov, I. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6491–6506 (2021).
30. Mitchell, E., Lin, C., Bosselut, A., Manning, C. D. & Finn, C. Memory-based model editing at scale. *International Conference on Machine Learning* **162**, 15817–15831 (2022).
31. Chen, Q. et al. Lifelong knowledge editing for LLMs with retrieval-augmented continuous prompt learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 13565–13580 (2024).
32. Tan, C., Zhang, G. & Fu, J. Massive editing for large language models via meta learning. In *The Twelfth International Conference on Learning Representations* (2024).
33. Tishby, N., Pereira, F. C. & Bialek, W. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, 368–377 (1999).
34. Alemi, A. A., Fischer, I., Dillon, J. V. & Murphy, K. Deep variational information bottleneck. In *5th International Conference on Learning Representations* (2017).

35. Hu, S., Lou, Z., Yan, X. & Ye, Y. A survey on information bottleneck. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 5325–5344 (2024).
36. Levy, O., Seo, M., Choi, E. & Zettlemoyer, L. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, 333–342 (2017).
37. Wang, R. & Li, P. Lemoe: Advanced mixture of experts adaptor for lifelong model editing of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2551–2575 (2024).
38. Hartvigsen, T., Sankaranarayanan, S., Palangi, H., Kim, Y. & Ghassemi, M. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems* **36**, 47934–47959 (2023).
39. Wang, P. et al. Easyedit: An easy-to-use knowledge editing framework for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 82–93 (2024).
40. Vrandecic, D. & Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Commun. ACM* **57**, 78–85 (2014).
41. Reimers, N. & Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 3980–3990 (2019).
42. Hu, C., Cao, P., Chen, Y., Liu, K. & Zhao, J. Wilke: Wise-layer knowledge editor for lifelong knowledge editing. In *Findings of the Association for Computational Linguistics*, 3476–3503 (2024).
43. Gupta, A., Rao, A. & Anumanchipalli, G. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics*, 15202–15232 (2024).
44. Yang, W. et al. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics*, 5419–5437 (2024).
45. Van der Maaten, L. & Hinton, G. Visualizing data using t-sne.J. Mach. Learn. Res. 9 (2008).
46. Zhang, T. et al. Dafnet: Dynamic auxiliary fusion for sequential model editing in large language models. In *Findings of the Association for Computational Linguistics*, 1588–1602 (2024).
47. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations* (2014).

Acknowledgements

This work was supported by the National Science and Technology Major Project (Grant No. 2022ZD0120302). In addition, we sincerely thank the Shanghai Institute of AI for Education (IAIE) for their computational power support.

Author contributions

Q.C. contributed to conceptualization, project planning, framework development, experimental design, software implementation, data processing and analysis, and manuscript writing. C.W. contributed to concept discussions, experimental design, data analysis, and manuscript revision. T.Z. participated in research discussions, data analysis, and oversaw experimental design and figure creation. X.H. contributed to conceptualization, project planning, and manuscript revision, and was responsible for overall communication and coordination. He managed manuscript submission and responses to reviewer feedback, and coordinated revisions. He also served as the primary contact for the author team and managed computational resources. X.H. supervised the work and is the corresponding author. All authors have read and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Xiaofeng He.