BRIDGING THE REPRESENTATION–BEHAVIOR GAP IN KNOWLEDGE EDITING WITH CALIBRATED ACTIVATION STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge editing aims to update specific knowledge in large language models while preserving unrelated behavior. Recent distribution-based approaches edit knowledge by steering hidden representations toward distributions associated with the target knowledge. However, it remains unclear whether better representation alignment necessarily leads to the desired output behavior. Through systematic layer-wise analysis, we identify a **representation–behavior gap** with two distinct failure modes: **behavioral under-realization**, where well-aligned hidden representations fail to sufficiently express the target knowledge in output behavior, and **behavioral overshooting**, where stronger interventions continue to improve edit success but push the output behavior beyond that induced by explicit conditioning on the target knowledge. To address these failure modes, we propose **B**ehavior-aware **E**diting via **C**alibrated **A**ctivation **S**teering (**BECAS**). BECAS uses the model's output distribution when explicitly conditioned on the target knowledge as a behavioral reference and adjusts the strength of activation steering accordingly. Experiments on knowledge editing benchmarks show that BECAS improves generalization and locality while maintaining high editing efficacy. These results demonstrate that output behavior provides a useful reference for controlling representation-based knowledge editing.

1 INTRODUCTION

Large language models (LLMs) encode extensive factual knowledge in their parameters, but this knowledge can become outdated or inconsistent with newly available information (Petroni et al., 2019; Brown et al., 2020; Lopez-Lira & Tang, 2026). Knowledge editing (KE) aims to selectively update such knowledge without retraining the model from scratch, while preserving unrelated knowledge and behavior (Sinitsin et al., 2020; Yao et al., 2023). A reliable editor should therefore achieve *efficacy* on the target fact, *generalization* to equivalent queries and relevant implications, and *locality* on unrelated inputs (Lazaridou et al., 2021; De Cao et al., 2021; Meng et al., 2022a).

Most existing knowledge editing methods formulate an edit using specific input–target pairs and modify model parameters or internal representations to increase the likelihood of the desired answer (Zhang et al., 2024; Meng et al., 2022b; Fang et al., 2025). Recent work has also explored set-based or distributional formulations of knowledge editing. Representation-based steering methods construct hidden-state distributions from paraphrases and logical implications and steer them toward distributions associated with the target knowledge (Turner et al., 2024; Scialanga et al., 2025; Zhao et al., 2025). In a complementary direction, self-induced editing uses the model's output distribution when explicitly conditioned on the target fact as a richer behavioral signal than a one-hot target (Qi et al., 2025). These approaches provide two complementary distributional views of an edit: one in representation space and the other in output behavior.

This raises a natural question: does better alignment with the target representation distribution necessarily lead to output behavior that more closely resembles that of a model conditioned on the target knowledge? We investigate this question through a systematic layer-wise analysis of distribution-based knowledge editing. Our analysis reveals a **representation–behavior gap**: representation alignment alone does not determine whether an intervention induces an appropriate degree of be-

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.

havioral change. We observe two distinct manifestations of this gap. First, **behavioral under-realization** occurs when hidden-state distributions are well aligned with the target distribution, while the target knowledge remains insufficiently expressed in the model's output behavior. Second, **behavioral overshooting** occurs when increasing the intervention strength continues to improve edit success but moves the output behavior beyond that induced by explicit conditioning on the target knowledge.

Based on these observations, we propose **B**ehavior-aware **E**diting via **C**alibrated **A**ctivation **S**teering (**BECAS**). BECAS uses the output distribution induced by explicit conditioning on the target knowledge as a behavioral reference for activation steering. Rather than fully applying a given representation intervention, BECAS adjusts its strength to sufficiently realize the target knowledge while avoiding behavioral overshooting.

This perspective also provides a simple interpretation of the efficacy–generalization–locality trade-off. An insufficient intervention may fail to realize the target knowledge in output behavior, reducing efficacy and generalization, whereas an excessive intervention may unnecessarily alter nearby behavior and degrade locality. BECAS seeks an intermediate intervention that realizes the target knowledge while limiting such unintended changes.

Our contributions are threefold:

- We identify a **representation–behavior gap** in knowledge editing through systematic layer-wise and instance-level analyses. We show that representation alignment does not necessarily lead to the desired output behavior, revealing two failure modes: behavioral under-realization and behavioral overshooting.
- We characterize the desired behavior of an edit using the model's output distribution when explicitly conditioned on the target knowledge. This behavioral reference allows us to assess whether a representation intervention is sufficient to realize the target knowledge without unnecessarily altering the model's output behavior.
- We propose **BECAS**, which uses the behavioral reference to adjust the strength of activation steering. Experiments on knowledge editing benchmarks show that BECAS improves generalization and locality while maintaining high editing efficacy, compared with existing editing methods.

2 Related Work

Parameter-Modifying Knowledge Editing. Parameter-modifying methods directly update model parameters to encode new factual associations (Zhang et al., 2024). Locate-and-edit methods identify parameters associated with factual recall: ROME (Meng et al., 2022b) applies rank-one updates to feed-forward layers, MEMIT (Meng et al., 2022c) extends this approach to multiple edits, and AlphaEdit (Fang et al., 2025) constrains updates to reduce interference with preserved knowledge. Other methods learn parameter updates, including KnowledgeEditor (De Cao et al., 2021) with auxiliary networks and MEND (Mitchell et al., 2021) with learned gradient transformations.

Augmentation-Based Knowledge Editing. Augmentation-based methods store or provide edited knowledge without directly modifying the original model parameters. SERAC (Mitchell et al., 2022) stores edits externally and routes relevant queries, while GRACE (Hartvigsen et al., 2023) uses an external codebook for replacement representations. In-context methods instead provide new knowledge through the input: IKE (Zheng et al., 2023) constructs demonstrations for a target edit, while ICE (Qi et al., 2025) uses the model's output distribution conditioned on the new knowledge as a self-induced editing target.

Activation-Based Knowledge Editing. Activation-based methods modify intermediate representations while leaving the pretrained parameters unchanged (Sun et al., 2026). Activation steering constructs directions from internal representations and applies them to hidden states (Turner et al., 2024). SAKE (Scialanga et al., 2025) extends this approach to distributional knowledge editing by transporting source activation distributions toward target distributions using Optimal Transport. SPARE (Zhao et al., 2025) uses sparse autoencoder features to manipulate representations associated

with knowledge-selection behavior. Our work studies whether alignment with a target representation distribution also yields the corresponding output behavior.

3 Representation and Behavioral Views of Knowledge Editing

3.1 Representation-Space Editing

Consider a pretrained language model f_θ and a knowledge edit $e = (s, r, o \to o^*)$, which replaces an existing association (s, r, o) with a target association (s, r, o^*). Let $\mathcal{Q}_e = \{q_i\}_{i=1}^N$ denote a set of queries associated with the edited knowledge, including paraphrases and relevant implications.

Following distributional activation-based knowledge editing (Scialanga et al., 2025), we characterize an edit using the hidden representations induced by these queries. At layer l, let $P^l_{S,e}(\boldsymbol{H}^l)$ and $P^l_{T,e}(\boldsymbol{H}^l)$ denote the source and target representation distributions, respectively. A representation intervention $T_l : \mathbb{R}^d \to \mathbb{R}^d$ transports the source distribution toward the target distribution:

$$\hat{P}^l_{T,e}(\boldsymbol{H}^l) = T_{l\#} P^l_{S,e}(\boldsymbol{H}^l), \tag{1}$$

where $T_{l\#}P$ denotes the pushforward of distribution P through T_l. Unless otherwise specified, we instantiate T_l using Gaussian-to-Gaussian optimal transport, following SAKE (Scialanga et al., 2025). We measure representation alignment by

$$D^l_H(e) = \mathcal{D}_H \left(\hat{P}^l_{T,e}(\boldsymbol{H}^l), P^l_{T,e}(\boldsymbol{H}^l) \right), \tag{2}$$

where $\mathcal{D}_H$ denotes a **representation-space discrepancy measure**.

3.2 Knowledge-Conditioned Behavioral Reference

To characterize the output behavior associated with the target knowledge, we provide the target fact to the pretrained model through a conditioning context c_e and define

$$P_{\mathrm{ctx},e}(Y \mid q) = P_\theta(Y \mid c_e, q). \tag{3}$$

This distribution serves as a model-intrinsic reference for the behavior induced when the target knowledge is explicitly available.

For an intervention T_l, let $P^l_{E,e}(Y \mid q)$ denote the output distribution of the intervened model. We define its behavioral discrepancy as

$$D^l_Y(e) = \mathbb{E}_{q \sim \mathcal{Q}_e} \left[\mathcal{D}_Y \left(P^l_{E,e}(Y \mid q), P_{\mathrm{ctx},e}(Y \mid q) \right) \right], \tag{4}$$

where $\mathcal{D}_Y$ denotes a **output-space discrepancy measure**.

3.3 From Representation Alignment to Output Behavior

An intervention at layer l is propagated through the remaining layers of the model. Let $g_{l:L}$ denote the computation from layer l to the output. The intervention therefore induces

$$P^l_{S,e}(\boldsymbol{H}^l) \xrightarrow{T_l} \hat{P}^l_{T,e}(\boldsymbol{H}^l) \xrightarrow{g_{l:L}} P^l_{E,e}(Y). \tag{5}$$

Representation-based editing directly controls the intervention in representation space, whereas its effect is ultimately observed in the output distribution. This raises the question of whether improved representation alignment is accompanied by improved behavioral alignment:

$$D^l_H(e) \downarrow \quad \stackrel{?}{\Longrightarrow} \quad D^l_Y(e) \downarrow . \tag{6}$$

A related question concerns intervention strength. Even when a stronger intervention increases the probability of the target answer, it may not necessarily move the output distribution closer to the knowledge-conditioned behavioral reference. We investigate these two questions empirically in Section 4.

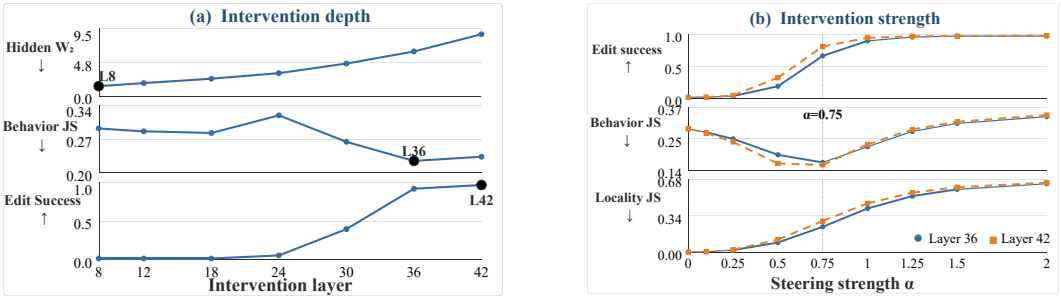

Figure 1: The representation–behavior gap across intervention depth and strength.

4 Representation–Behavior Gap in Knowledge Editing

Section 3 introduced two views of a representation intervention: its alignment with the target representation distribution and its alignment with the knowledge-conditioned behavioral reference. We now examine their empirical relationship. Specifically, we ask two questions: (i) whether better representation alignment leads to better behavioral alignment, and (ii) whether higher edit success indicates better behavioral alignment.

Diagnostic setup. We use SAKE (Scialanga et al., 2025) as a representative distributional activation-based editor and analyze interventions across transformer layers on CounterFact. For the representation-space discrepancy $\mathcal{D}_H$, we use the 2-Wasserstein distance between the transported and target representation distributions (Representation W_2). For the output-space discrepancy $\mathcal{D}_Y$, we use the Jensen–Shannon (JS) divergence between the edited output distribution and the knowledge-conditioned behavioral reference (Behavior JS). We additionally report edit success (Eff.) and the JS divergence between the edited and original output distributions on neighborhood queries (Locality JS).

For Behavior JS, we compare next-token distributions. Let $\Pi_K(P, Q)$ denote the renormalized projection of P and Q onto the union of their top-K tokens. For an edit e and query q, we compute

$$D_Y^l(e) = \mathbb{E}_{q\sim\mathcal{Q}_e}\left[D_{\mathrm{JS}}\left(\Pi_K\left(P_{E,e}^l(Y_1 \mid q), P_{\mathrm{ctx},e}(Y_1 \mid q)\right)\right)\right], \tag{7}$$

where $K = 100$ throughout. The detailed definition of Π_K is provided in Appendix A.1. Locality JS is computed analogously between the edited and original model distributions on neighborhood queries. All analyses in this section are conducted on a diagnostic development set that is disjoint from the final evaluation instances in Section 6. We use this set only to analyze intervention depth and strength and to determine the candidate intervention layers used in subsequent experiments.

4.1 Behavioral Under-Realization despite Representation Alignment

We first examine whether better representation alignment is accompanied by better behavioral alignment, as posed in Eq. 6. If the two were closely related, layers with smaller $D_H^l(e)$ would also tend to exhibit smaller $D_Y^l(e)$.

Figure 1(a) shows that the two discrepancies are minimized at substantially different layers. Representation alignment is strongest at layer 8, with Representation W_2 of 1.36, yet the edit success at this layer is only 0.8%. In contrast, Behavior JS is minimized at layer 36, where edit success reaches 91.6%, while the highest edit success of 96.5% is attained at layer 42. Thus, the representation-, behavior-, and efficacy-optimal layers are $l^*_{\mathrm{repr}} = 8$, $l^*_{\mathrm{beh}} = 36$, and $l^*_{\mathrm{eff}} = 42$, respectively.

Finding 1: Behavioral Under-Realization. *Strong representation alignment can coexist with insufficient realization of the target knowledge in output behavior.*

4.2 Edit Success and Behavioral Overshooting

We next examine whether edit success provides a sufficient criterion for determining intervention strength. To separate intervention strength from intervention depth, we fix the representation trans-

port T_l and vary its strength as

$$\tilde{\boldsymbol{h}}_{\alpha}^{l} = \boldsymbol{h}^{l} + \alpha \left(T_l(\boldsymbol{h}^{l}) - \boldsymbol{h}^{l} \right), \tag{8}$$

where $\alpha = 0$ leaves the representation unchanged and $\alpha = 1$ applies the full transport.

Figure 1(b) shows that the strength minimizing behavioral discrepancy can differ from that yielding higher edit success. At both layers 36 and 42, BEHAVIOR JS is minimized at $\alpha = 0.75$, while edit success continues to increase beyond this point. At layer 36, increasing α from 0.75 to 1.0 raises edit success from 66.7% to 90.3%, while BEHAVIOR JS increases from 0.163 to 0.218. At layer 42, edit success similarly increases from 79.5% to 94.7%, while BEHAVIOR JS increases from 0.154 to 0.226. We refer to this pattern as **behavioral overshooting**: for two strengths $\alpha_1 < \alpha_2$, edit success increases, $\mathrm{Eff}(\alpha_2) > \mathrm{Eff}(\alpha_1)$, while behavioral discrepancy also increases, $D_Y^{l,\alpha_2}(e) > D_Y^{l,\alpha_1}(e)$. Here, $D_Y^{l,\alpha}(e)$ denotes the behavioral discrepancy at layer l with intervention strength α.

The same increase in intervention strength also produces larger changes on neighborhood queries. As α increases from 0.75 to 1.0, LOCALITY JS increases from 0.231 to 0.400 at layer 36 and from 0.279 to 0.447 at layer 42. Thus, the additional intervention that improves edit success is accompanied by greater behavioral discrepancy and larger changes to unrelated behavior.

Finding 2: Behavioral Overshooting. *Increasing intervention strength can improve edit success while moving the output behavior away from the knowledge-conditioned reference.*

4.3 THE REPRESENTATION–BEHAVIOR GAP

Together, the two findings reveal a **representation–behavior gap**: neither representation alignment nor edit success alone determines how closely an intervention reproduces the knowledge-conditioned behavioral reference. Behavioral under-realization shows that strong representation alignment can be insufficient, whereas behavioral overshooting shows that stronger intervention can become excessive even as edit success continues to improve.

These findings suggest that an appropriate representation intervention should account for its induced output behavior. This motivates using the knowledge-conditioned behavioral distribution as a reference for controlling intervention strength, which we develop in Section 5.

5 BECAS: BEHAVIOR-AWARE EDITING VIA CALIBRATED ACTIVATION STEERING

Section 4 showed that representation interventions can exhibit both behavioral under-realization and behavioral overshooting. These findings suggest that representation transport alone does not determine an appropriate intervention strength. We therefore propose **B**ehavior-aware **E**diting via **C**alibrated **A**ctivation **S**teering (**BECAS**), which uses knowledge-conditioned output behavior as a reference for controlling representation interventions. BECAS consists of two components: (i) constructing a reliable behavioral reference from multiple knowledge-conditioning contexts, and (ii) selecting an intervention that balances behavioral alignment, sufficient edit efficacy, and locality.

5.1 RELIABLE KNOWLEDGE-CONDITIONED BEHAVIORAL REFERENCE

Section 3.2 defined the behavioral reference as the output distribution induced when the target knowledge is explicitly provided. Since this distribution may vary with the conditioning context, we construct the reference from multiple contexts and assess its reliability before using it for calibration.

Given an edit $e = (s, r, o \to o^*)$ and a query q, let $\{c_e^{(k)}\}_{k=1}^{K}$ denote K contexts expressing the target fact. We define the aggregated behavioral reference as

$$Q_{e,q}^{\mathrm{ctx}}(Y) = \frac{1}{K} \sum_{k=1}^{K} P_\theta(Y \mid c_e^{(k)}, q). \tag{9}$$

A useful behavioral reference should satisfy two properties: *context agreement*, requiring different contexts expressing the same fact to induce similar output distributions, and *target support*, requiring these distributions to assign sufficient probability to o^*. We quantify context agreement by the average pairwise JS divergence

$$d_{\mathrm{ctx}}(q) = \frac{2}{K(K-1)} \sum_{1 \le i < j \le K} D_{\mathrm{JS}} \left(P_\theta(Y \mid c_e^{(i)}, q), P_\theta(Y \mid c_e^{(j)}, q) \right), \tag{10}$$

and target support by

$$p_{\mathrm{new}}^{\mathrm{ctx}}(q) = \frac{1}{K} \sum_{k=1}^{K} P_\theta(o^* \mid c_e^{(k)}, q). \tag{11}$$

A small $d_{\mathrm{ctx}}(q)$ indicates agreement across contexts, while a large $p_{\mathrm{new}}^{\mathrm{ctx}}(q)$ indicates that the target knowledge is expressed in the induced behavior.

We combine these two quantities into the reliability weight

$$r_q = \mathbb{I}\left[d_{\mathrm{ctx}}(q) \le \tau_{\mathrm{JS}},\ p_{\mathrm{new}}^{\mathrm{ctx}}(q) \ge \tau_p \right] p_{\mathrm{new}}^{\mathrm{ctx}}(q) \left(1 - d_{\mathrm{ctx}}(q)\right). \tag{12}$$

The indicator excludes references with low context agreement or insufficient target support, while the remaining terms assign larger weights to references with stronger agreement and target support.

Finally, we define the behavioral alignment loss as

$$\mathcal{L}_{\mathrm{beh}}(e, l, \alpha) = \frac{\sum_{q \in \mathcal{Q}_{\mathrm{cal}}^{\mathrm{beh}}} r_q D_{\mathrm{JS}} \left(P_{E,e}^{l,\alpha}(Y \mid q), Q_{e,q}^{\mathrm{ctx}}(Y) \right)}{\sum_{q \in \mathcal{Q}_{\mathrm{cal}}^{\mathrm{beh}}} r_q}. \tag{13}$$

Thus, behavioral calibration is driven primarily by queries for which the knowledge-conditioned reference is both stable across contexts and supportive of the target knowledge. If $\sum_q r_q = 0$, we set $\mathcal{L}_{\mathrm{beh}} = 0$.

5.2 Behavior-Calibrated Activation Steering

Let T_l denote the representation transport defined in Section 3.1. Rather than applying the full transport, BECAS controls its strength as

$$\tilde{\boldsymbol{h}}_\alpha^l = \boldsymbol{h}^l + \alpha \left(T_l(\boldsymbol{h}^l) - \boldsymbol{h}^l \right), \qquad \alpha \in [\alpha_{\min}, \alpha_{\max}]. \tag{14}$$

Here, $T_l(\boldsymbol{h}^l) - \boldsymbol{h}^l$ determines the steering direction, while α controls the intervention strength. The cases $\alpha = 0$ and $\alpha = 1$ correspond to no intervention and the full representation transport, respectively.

Sufficient efficacy. Minimizing behavioral discrepancy alone does not ensure that the target knowledge is sufficiently realized. To prevent behavioral under-realization, we measure efficacy using the length-normalized log-probability margin between the target and original objects:

$$S_{\mathrm{eff}}(e, l, \alpha) = \mathbb{E}_{q \in \mathcal{Q}_{\mathrm{cal}}^{\mathrm{beh}}} \left[\sigma \left(\bar{\ell}_{l,\alpha}(o^* \mid q) - \bar{\ell}_{l,\alpha}(o \mid q) \right) \right], \tag{15}$$

where $\bar{\ell}_{l,\alpha}(o \mid q)$ denotes the length-normalized log-probability of object o under the intervention at layer l with strength α. We then define the sufficient-efficacy loss as

$$\mathcal{L}_{\mathrm{eff}}(e, l, \alpha) = \left[\tau - S_{\mathrm{eff}}(e, l, \alpha) \right]_+. \tag{16}$$

This loss penalizes interventions with insufficient efficacy but becomes zero once $S_{\mathrm{eff}}(e, l, \alpha) \ge \tau$. It therefore provides no additional incentive to increase the intervention strength after the required efficacy level is reached.

Locality preservation. Section 4.2 showed that stronger interventions can increase changes to unrelated behavior. We therefore measure locality on calibration neighborhood queries and define

$$\mathcal{L}_{\mathrm{loc}}(e, l, \alpha) = \mathbb{E}_{q \in \mathcal{Q}_{\mathrm{cal}}^{\mathrm{loc}}} \left[D_{\mathrm{JS}} \left(P_{E,e}^{l,\alpha}(Y \mid q), P_\theta(Y \mid q) \right) \right]. \tag{17}$$

This term penalizes interventions that alter the output behavior on queries unrelated to the target edit.

Overall objective. Combining behavioral alignment, sufficient efficacy, and locality gives

$$\mathcal{L}_{\text{BECAS}}(e,l,\alpha) = \mathcal{L}_{\text{beh}}(e,l,\alpha) + \lambda\mathcal{L}_{\text{loc}}(e,l,\alpha) + \gamma\mathcal{L}_{\text{eff}}(e,l,\alpha), \tag{18}$$

where λ and γ control the contributions of locality and efficacy, respectively.

For each candidate layer $l \in \mathcal{L}$, we first select the steering strength

$$\alpha^*_{e,l} = \arg\min_{\alpha\in[\alpha_{\min},\alpha_{\max}]} \mathcal{L}_{\text{BECAS}}(e,l,\alpha). \tag{19}$$

We then select the intervention layer as

$$l^*_e = \arg\min_{l\in\mathcal{L}} \mathcal{L}_{\text{BECAS}}(e,l,\alpha^*_{e,l}), \qquad \alpha^*_e = \alpha^*_{e,l^*_e}. \tag{20}$$

The candidate layer set $\mathcal{L}$ is determined using only the diagnostic development set in Section 4 and is fixed before final evaluation. The complete BECAS procedure is summarized in Algorithm 1, with implementation details provided in Section 6.1.

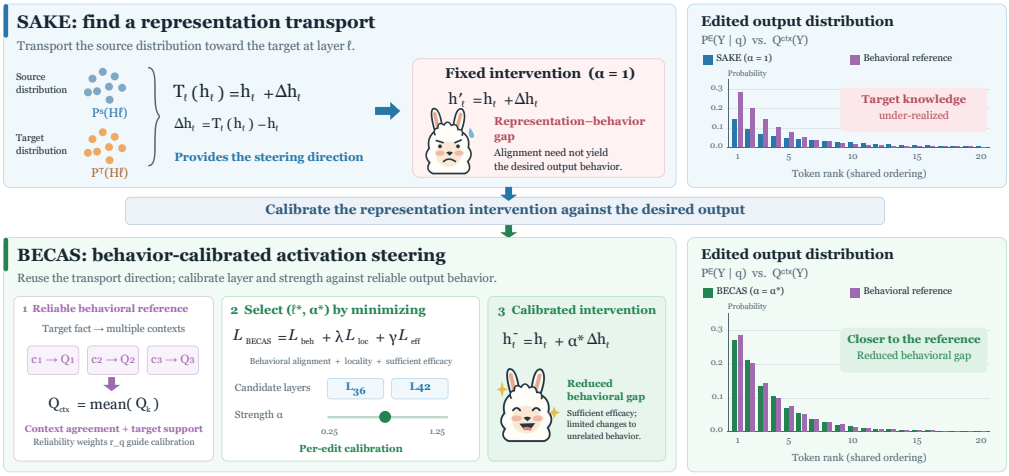

Figure 2: Overview of BECAS. BECAS uses an existing representation transport as the steering direction and a reliable knowledge-conditioned behavioral reference to select the intervention layer and calibrate its strength.

6 Experiments

6.1 Experimental Setup

Datasets. We evaluate BECAS on CounterFact (Meng et al., 2022b) and zsRE (Levy et al., 2017), two standard knowledge-editing benchmarks. We follow their standard evaluation protocols for efficacy, generalization, and locality. Additional dataset details are provided in Appendix A.3.

Baselines. We compare BECAS with parameter-modifying methods ROME (Meng et al., 2022b), MEMIT (Meng et al., 2022c), in-context editing method ICE (Qi et al., 2025) and activation-based methods ActAdd (Turner et al., 2024) and SAKE (Scialanga et al., 2025). SAKE is the closest baseline because BECAS uses its distributional representation transport and calibrates the resulting activation intervention rather than learning a new transport.

Evaluation metrics. We distinguish performance on the edit request from evaluation on held-out queries. Efficacy (**Eff.**) is measured on the original rewrite query, generalization (**Gen.**) on held-out paraphrase queries, and locality (**Loc.**) on neighborhood queries. We additionally report Behavior JS (**B.JS**), defined in Section 4, and behavioral overshoot (**OS**) on held-out paraphrase queries.

Behavioral overshoot measures the positive excess in target preference relative to the knowledge-conditioned behavioral reference:

$$\text{OS} = \mathbb{E}_{q\in\mathcal{Q}^{\text{in}}}\left[\left(s_{P_E}(q,o^*) - s_{Q^{\text{ctx}}}(q,o^*)\right)_+\right], \tag{21}$$

Table 1: Knowledge editing performance on COUNTERFACT and ZSRE with GPT2-XL. Higher is better for Eff., Gen., and Loc.; lower is better for B.JS and OS.

CounterFact					
Method	**Eff.**	**Gen.**	**Loc.**	**B.JS.↓**	**OS.↓**
MEMIT	61.12	36.67	67.32	34.88	31.63
ROME	**99.23**	73.26	81.75	30.16	45.24
ActAdd	90.36	30.52	82.58	32.17	30.27
SAKE	96.20	87.54	84.67	27.79	43.46
SAKE @$L36$	92.78	86.43	85.20	25.83	35.30
SAKE @$L42$	93.44	90.78	83.55	27.23	42.48
BECAS	93.37	**93.13**	**90.84**	**15.92**	**7.67**

ZsRE					
Method	Eff.	Gen.	Loc.	B.JS.↓	OS.↓
MEMIT	65.70	43.53	70.06	35.28	30.43
ROME	**98.75**	78.74	82.58	31.20	45.13
ActAdd	89.04	68.34	83.18	31.72	31.27
SAKE	95.47	86.27	87.31	28.08	42.13
SAKE @$L36$	92.19	86.60	87.23	25.26	32.72
SAKE @$L42$	94.63	89.09	84.27	27.90	40.43
BECAS	92.72	**92.01**	**90.32**	**10.82**	**11.24**

where $s_P(q, o^*)$ denotes the length-normalized score assigned to the target object o^* under distribution P. Lower OS indicates less amplification of the target object beyond the knowledge-conditioned reference.

Implementation. For each edit, BECAS uses a calibration set consisting of the rewrite, one sampled paraphrase, and one sampled neighborhood query for calibration. Based on Section 4, candidate layers are $\mathcal{L} = \{36, 42\}$, with $\alpha \in [0.25, 1.25]$. We set $\lambda = \gamma = 1$, $\tau = 0.8$, $\tau_{\text{JS}} = 0.1$, and $\tau_p = 0.2$. Further details are provided in Appendix A.3.

6.2 MAIN RESULTS

Table 1 compares BECAS with representative knowledge editors on COUNTERFACT. Compared with SAKE, BECAS reduces BEHAVIOR JS from 27.79 to 15.92 and improves locality from 84.67% to 90.84%, while maintaining 93.37% efficacy. These results indicate that BECAS produces output behavior closer to the knowledge-conditioned behavioral reference without sacrificing edit efficacy.

We observe a similar pattern on ZSRE. BECAS achieves 92.72% efficacy, 92.01% generalization, and 90.32% locality, with BEHAVIOR JS and behavioral overshoot (**OS**) of 10.82 and 11.24, respectively. Across both datasets, BECAS improves behavioral alignment and locality while maintaining high efficacy and generalization.

Performance by representation–behavior gap. We next examine whether the benefit of BECAS varies with the degree of representation–behavior mismatch identified in Section 4. For each of $1{,}000$ COUNTERFACT edits, we measure the pre-calibration gap at fixed operating point $l = 42$ and $\alpha = 1$ as $G_e = z\Big(D_Y^{42,1}(e)\Big) - z\big(D_H^{42}(e)\big)$, where $z(\cdot)$ denotes standardization across the analyzed edits. A larger G_e indicates that the behavioral discrepancy is high relative to the representation discrepancy. We divide the edits into Low, Medium, and High groups according to G_e.

Table 2 shows that the improvement from BECAS increases with G_e. Relative to the fixed intervention at $l = 42$ and $\alpha = 1$, the reduction in BEHAVIOR JS is 8.15, 9.89, and 18.94 for the Low, Medium, and High groups, respectively. Behavioral overshoot exhibits a similar pattern, with larger reductions for higher-gap edits. In contrast, locality improvements are concentrated in the High-gap group, where locality increases from 84.03 to 86.01. These results suggest that BECAS is particularly beneficial for edits with a large representation–behavior gap, consistent with the gap identified in Section 4.

6.3 ABLATION STUDY

We ablate BECAS on $1{,}000$ held-out CounterFact edits in table 3. Fixing either the steering strength or intervention layer increases behavioral discrepancy, supporting joint calibration of both dimensions.

The individual loss terms have distinct effects. Removing $\mathcal{L}_{\text{beh}}$ increases BEHAVIOR

Table 2: Performance across different levels of the representation–behavior gap.

Gap	**Method**	**Gen.**	**Loc.**	**B.JS↓**	**ΔB.JS↑**	**OS.↓**
Low	fixed point	89.80	68.00	20.78	–	33.77
	BECAS	87.29	64.55	12.64	8.15	4.98
Medium	fixed point	90.92	85.30	27.25	–	40.90
	BECAS	94.80	73.36	17.36	9.89	9.26
High	fixed point	89.02	84.03	35.51	–	59.37
	BECAS	93.92	86.01	16.57	18.94	17.81

JS from 16.92 to 23.30, while removing $\mathcal{L}_{\text{loc}}$ decreases locality from 89.13% to 82.16%. Removing the sufficient-efficacy term $\mathcal{L}_{\text{eff}}$ reduces efficacy from 92.78% to 90.39%. Finally, removing reliability weighting increases behavioral overshoot (**OS**) from 11.23 to 17.18. These results show that behavioral alignment, locality preservation, sufficient efficacy, and reliability weighting each contribute to the performance of BECAS.

6.4 Analysis

Instance-specific calibration. Figure 3 shows substantial variation in the selected steering strength and layer across 1,000 CounterFact edits (mean $\alpha = 0.72$; 57.5% select layer 42), supporting the need for instance-specific behavioral calibration.

Table 3: Ablation study of BECAS on CounterFact.

Method	Eff.	Gen	Loc.	B.JS	OS
Fixed α=1	94.25	90.68	84.57	26.32	41.34
Fixed α=0.75	89.21	88.73	87.62	19.57	20.18
Fixed l+continuous α	92.14	91.79	89.18	17.04	12.43
BECAS w/o reliability	91.75	90.34	88.17	18.50	17.18
BECAS w/o L_{eff}	90.39	90.22	89.15	18.73	15.64
BECAS w/o L_{beh}	93.13	89.93	87.88	23.30	28.95
BECAS w/o L_{loc}	93.69	91.56	82.16	19.97	13.71
full **BECAS**	92.78	**93.16**	**89.13**	**16.92**	**11.23**

The variation in both intervention layer and strength indicates that the behaviorally preferred intervention is not identical across edits, consistent with the instance-level representation–behavior mismatch observed in Section 4.

Reliability of behavioral references. The knowledge-conditioned behavioral reference is available for most edits: 954 of 1,000 edits (95.4%) contain at least one behavioral query that satisfies both the context-agreement and target-support criteria. For the remaining 46 edits (4.6%), BECAS excludes the behavioral reference and selects the intervention using only the efficacy and locality terms. Sensitivity to the reliability thresholds and context templates is reported in Appendix B.

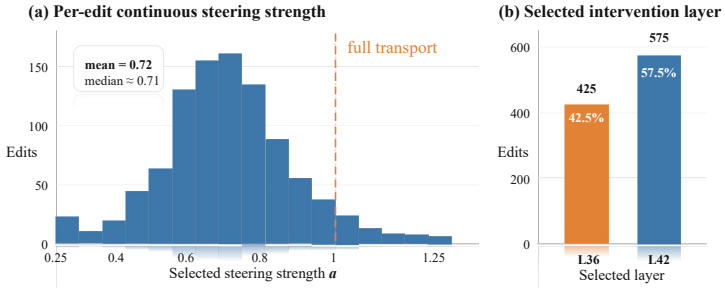

Figure 3: Instance-specific calibration in BECAS. (a) Distribution of selected steering strengths α. (b) Distribution of selected intervention layers.

7 Conclusion

In this work, we study the relationship between representation alignment and output behavior in distribution-based knowledge editing. Our analysis identifies a **representation–behavior gap** with two distinct manifestations: **behavioral under-realization**, where well-aligned representations fail to sufficiently realize the target knowledge in output behavior, and **behavioral overshooting**, where stronger interventions improve edit success while moving the output behavior away from the knowledge-conditioned behavioral reference.

Based on these findings, we introduce **BECAS**, which uses reliable knowledge-conditioned output distributions as behavioral references for calibrating activation steering. BECAS controls intervention strength while maintaining sufficient edit efficacy and limiting changes to unrelated behavior. Experiments on knowledge editing benchmarks show that BECAS improves behavioral alignment and locality while maintaining high efficacy and generalization. These results suggest that output behavior provides a useful reference for controlling representation-based knowledge editing.

Ethical Considerations

Our study involves experiments utilizing publicly accessible large language models, along with publicly available benchmark datasets CounterFact that have been widely employed and validated

in prior research. These models and datasets have been carefully curated and published by their original authors to mitigate potential ethical concerns such as biases, harmful outputs, and privacy risks.

Reproducibility Statement

We place strong emphasis on the transparency and reproducibility of our work. Section 6 of the main text outlines the experimental pipeline, including dataset preparation, prompts/context we used and training procedures. For further clarity, Appendix A.3 documents the full set of hyperparameter choices and auxiliary details. Together, these resources ensure that our results can be reliably replicated and extended in future research.

AI Use Statement

Generative AI tools were used to assist with language polishing, clarity improvements, and minor coding and LaTeX support. All research ideas, methodological design, experimental results, analyses, and conclusions were developed and verified by the authors.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6491–6506, 2021.

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.

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.

Angeliki Lazaridou, Adhi Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Tomas Kocisky, Sebastian Ruder, et al. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34:29348–29363, 2021.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, pp. 333–342, 2017.

Alejandro Lopez-Lira and Yuehua Tang. Can chatgpt forecast stock price movements? return predictability and large language models. *Journal of Financial Economics*, 184:104335, 2026.

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, 2022a.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Advances in Neural Information Processing Systems*, 2022b.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022c.

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

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International conference on machine learning*, pp. 15817–15831. PMLR, 2022.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2463–2473, 2019.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. In *International Conference on Learning Representations*, volume 2025, pp. 77563–77585, 2025.

Marco Scialanga, Thibault Laugel, Vincent Grari, and Marcin Detyniecki. Sake: Steering activations for knowledge editing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15966–15978, 2025.

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

Chung-En Sun, Ge Yan, Zimo Wang, and Tsui-Wei Weng. Steer2edit: From activation steering to component-level editing. *arXiv preprint arXiv:2602.09870*, 2026.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. 2024.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Steering knowledge selection behaviours in llms via sae-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5117–5136, 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.

Algorithm 1 Behavior-Aware Editing via Calibrated Activation Steering (BECAS)

Require: Edit $e = (s, r, o \to o^*)$; candidate layers $\mathcal{L}$; transports $\{T_l\}_{l\in\mathcal{L}}$; calibration queries $\mathcal{Q}_{\text{cal}}^{\text{beh}}$, $\mathcal{Q}_{\text{cal}}^{\text{loc}}$; contexts $\{c_e^{(k)}\}_{k=1}^{K}$
Ensure: Selected intervention (l_e^*, α_e^*)

```
for q ∈ Q_cal^beh do
    Q_{e,q}^{(k)} ← P_θ(Y | c_e^{(k)}, q) for k = 1, ..., K
    Q_{e,q}^{ctx} ← K^{-1} Σ_{k=1}^{K} Q_{e,q}^{(k)}
    Compute d_ctx(q) and p_new^{ctx}(q) using Eqs. 10–11
    Compute reliability r_q using Eq. 12
end for
for l ∈ L do
    Define L_BECAS(e, l, α) using Eq.18,
    α*_{e,l} ← arg min_{α∈[α_min,α_max]} L_BECAS(e, l, α)
end for
l*_e ← arg min_{l∈L} L_BECAS(e, l, α*_{e,l})
α*_e ← α*_{e,l*_e}
return (l*_e, α*_e)
```

A Additional Method Details

A.1 Details of Behavioral Distribution Comparison

Top-K projection. We compute behavioral discrepancy over a shared token support to avoid comparing the full vocabulary distribution. Given two next-token distributions P and Q, let

$$\mathcal{V}_K(P, Q) = \mathrm{TopK}(P) \cup \mathrm{TopK}(Q), \tag{22}$$

where $\mathrm{TopK}(P)$ denotes the set of K tokens with the highest probability under P.

For any distribution $R \in \{P, Q\}$, its renormalized restriction to the shared support is defined as

$$(\Pi_{\mathcal{V}_K} R)(y) = \frac{R(y)\mathbf{1}[y \in \mathcal{V}_K(P, Q)]}{\sum_{y' \in \mathcal{V}_K(P,Q)} R(y')}. \tag{23}$$

We then define

$$\Pi_K(P, Q) = (\Pi_{\mathcal{V}_K} P,\, \Pi_{\mathcal{V}_K} Q)\,. \tag{24}$$

Accordingly, the behavioral discrepancy for an edit e at layer l is

$$D_Y^l(e) = \mathbb{E}_{q\sim\mathcal{Q}_e}\left[D_{\mathrm{JS}}\left(\Pi_K\left(P_{E,e}^l(Y_1 \mid q), \mathrm{Q}_{e,q}^{ctx}(Y_1 \mid q)\right)\right)\right]. \tag{25}$$

We use $K = 100$ in all experiments. For Locality JS, we apply the same procedure to the next-token distributions of the edited and original models on neighborhood queries. We use next-token distributions to provide a common token-level behavioral comparison across queries and models.

A.2 BECAS Algorithm

Algorithm 1 summarizes BECAS for a single knowledge edit. The representation transports $\{T_l\}_{l\in\mathcal{L}}$ are provided by an existing distributional representation editor; BECAS does not learn a new transport. Instead, it constructs a reliable knowledge-conditioned behavioral reference and uses it to select the intervention layer and steering strength.

A.3 Additional Implementation

Dataset and calibration split. For each edit, BECAS uses separate queries for calibration and evaluation. The calibration set consists of the rewrite query, one sampled paraphrase query, and one sampled neighborhood query. The remaining paraphrase, generation, and neighborhood queries are reserved for evaluation. This split is performed independently for each edit, and no held-out evaluation query is used for calibration.

Candidate-layer protocol. The candidate intervention layers are determined once using the diagnostic development set in Section 4, which is disjoint from the final evaluation instances. The layer-wise analysis identifies two complementary operating points: layer 36 minimizes behavioral discrepancy, whereas layer 42 yields higher edit efficacy. We therefore fix $\mathcal{L} = \{36, 42\}$ for all main experiments. BECAS performs instance-specific selection only within this fixed candidate set.

Hyperparameters. Unless otherwise specified, we use GPT2-XL with candidate layers $\mathcal{L} = \{36, 42\}$ and optimize the steering strength over $\alpha \in [0.25, 1.25]$ using bounded scalar search with eight iterations. We set $\lambda = 1$, $\gamma = 1$, the efficacy threshold $\tau = 0.8$, the context-agreement threshold $\tau_{\text{JS}} = 0.1$, and the minimum target-support threshold $\tau_p = 0.2$.

B Additional Analysis

B.1 Representation–Behavior Mismatch and Editing Failure

We further examine the relationship between the representation–behavior gap and edit success. For each edit, we compare the representation discrepancy and behavioral discrepancy induced by the same intervention and group the instances according to conventional target-answer success. As shown in Figure 4, failed edits exhibit larger behavioral discrepancy than successful edits even when representation discrepancy is small. This pattern is consistent with behavioral under-realization: close alignment with the target representation distribution does not necessarily result in successful realization of the target knowledge in output behavior.

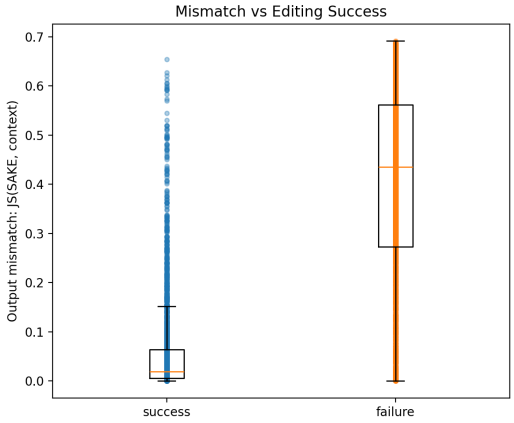

Figure 4: Representation and behavioral discrepancies for successful and failed edits. Failed edits can exhibit low representation discrepancy while retaining relatively high behavioral discrepancy.

B.2 Behavioral Overshooting and Locality

We next examine the relationship between behavioral overshooting and changes to unrelated behavior. Figure 5 compares behavioral overshoot with the distribution shift on neighborhood queries. Larger overshoot is associated with larger neighborhood distribution shifts, indicating that excessive target amplification tends to coincide with greater changes to unrelated behavior. This observation is consistent with the locality term in BECAS, which discourages interventions that improve target behavior at the expense of unrelated model behavior.

B.3 Additional Calibration Analysis

B.3.1 Layer-wise Representation–Behavior Analysis

Figure 6 provides the full layer-wise analysis used to determine the candidate intervention layers for BECAS. As shown in Figure 6(a), edit, paraphrase, and generation success remain low at earlier

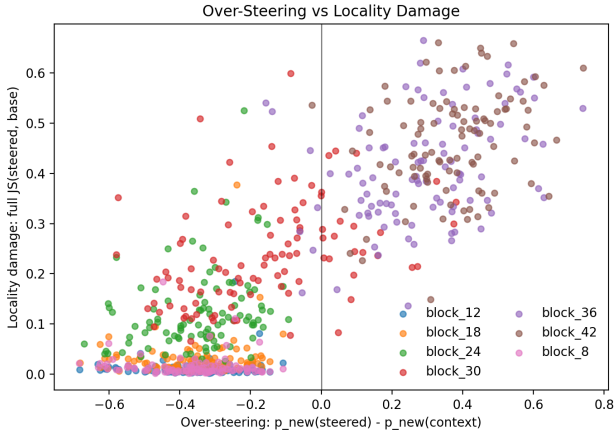

Figure 5: Relationship between behavioral overshoot and locality change. Larger behavioral overshoot is associated with greater distribution shift on neighborhood queries.

layers and increase at later layers, while neighborhood preservation generally decreases. Intervention depth therefore affects both realization of the target knowledge and preservation of unrelated behavior.

Figure 6(b) compares representation and behavioral discrepancies across layers. The two discrepancies follow different layer-wise patterns and attain their minima at different depths. In particular, strong representation alignment at earlier layers does not necessarily yield corresponding behavioral alignment, while later interventions can improve target realization with larger changes to neighborhood behavior. These results provide additional layer-wise evidence for the representation–behavior gap analyzed in Section 4.

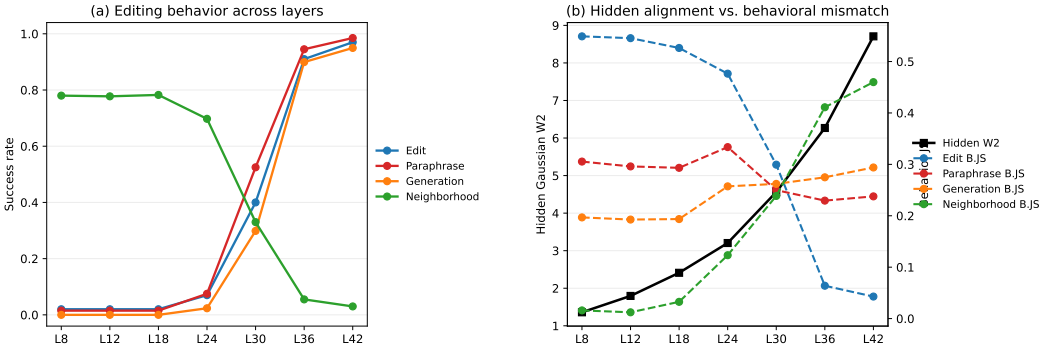

Figure 6: Layer-wise analysis on the diagnostic development set. (a) Edit, paraphrase, and generation success generally increase at later layers, while neighborhood preservation decreases. (b) Representation and behavioral discrepancies exhibit different layer-wise patterns and attain their minima at different intervention depths.

B.3.2 Steering-Strength Sensitivity

Figure 7 shows the effect of varying the intervention strength α. Increasing α improves in-scope success, but behavioral discrepancy is non-monotonic: Behavior JS first decreases and then increases after reaching an intermediate minimum. At the same time, stronger interventions produce larger changes on neighborhood queries. Thus, the strength that maximizes edit success need not minimize behavioral discrepancy, consistent with the behavioral overshooting observed in Section 4.2. This result motivates calibrating the intervention strength for each edit rather than using a fixed global value.

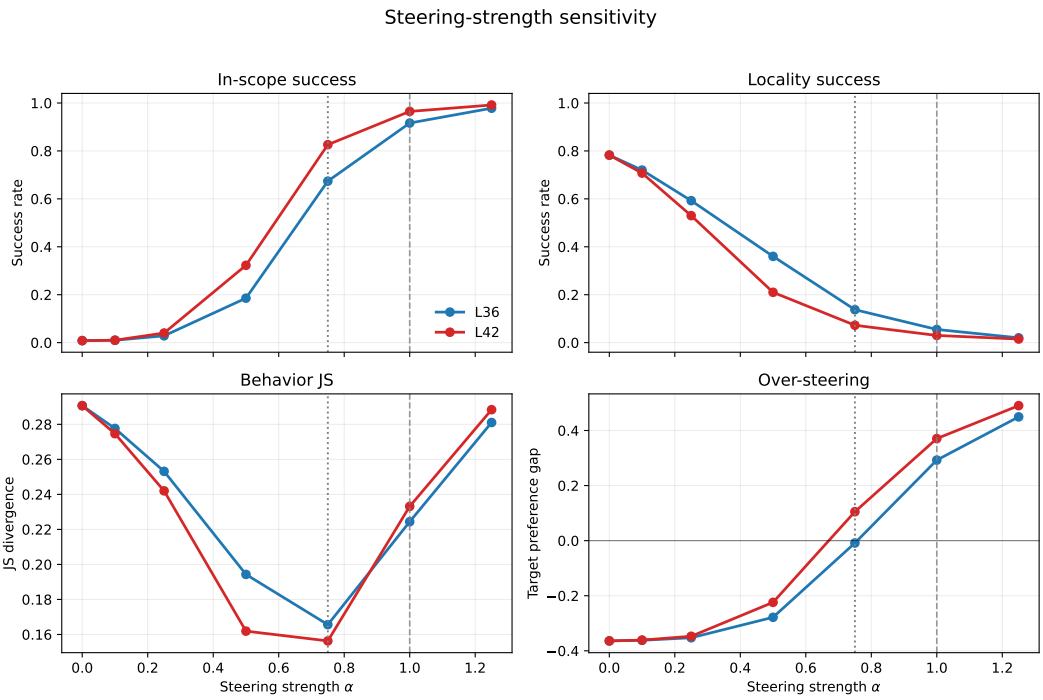

Figure 7: Sensitivity to intervention strength on the diagnostic development set. Increasing α improves in-scope success, whereas BEHAVIOR JS reaches its minimum at an intermediate strength and increases with further steering. Stronger interventions are also associated with larger changes on neighborhood queries. The dotted and dashed vertical lines indicate $\alpha = 0.75$ and $\alpha = 1.0$, respectively.

B.4 RELIABILITY ANALYSIS

We examine how often the knowledge-conditioned behavioral reference satisfies the reliability criteria defined in Section 5.1. On the full COUNTERFACT evaluation, 20,921 of 21,919 edits (95.4%) contain at least one behavioral query satisfying both the context-agreement and target-support criteria. On the 1,000-edit subset used for detailed analysis, this condition holds for 954 edits (95.4%). For an edit with no reliable behavioral query, BECAS sets $\mathcal{L}_{\text{beh}} = 0$ and selects the intervention using only the efficacy and locality terms.

Context templates. For each edit $e = (s, r, o \to o^*)$, we construct multiple contexts that explicitly provide the target fact. Let $f_e =$ `[rewrite prompt]` o^* denote a textual statement of the target fact. Table 4 lists the three templates used in our experiments.

Table 4: Context templates used to construct the knowledge-conditioned behavioral reference.

Template	**Context** $c_e^{(k)}(q)$
ICE-fact	*The following updated fact is true:* f_e *Complete this sentence:* q
Assume	*Assume this new fact is correct:* f_e *Now complete:* q
Updated-KB	*In the updated knowledge base,* f_e *Complete the query using the updated fact:* q

Each template induces a context-conditioned output distribution $P_\theta(Y \mid c_e^{(k)}, q)$. As defined in Section 5.1, BECAS aggregates these distributions into $Q_{e,q}^{\text{ctx}}$ and evaluates their reliability using context agreement $d_{\text{ctx}}(q)$ and target support $p_{\text{new}}^{\text{ctx}}(q)$. A query is retained when $d_{\text{ctx}}(q) \le \tau_{\text{JS}}$ and $p_{\text{new}}^{\text{ctx}}(q) \ge \tau_p$.

Table 5 reports these statistics for different query types. Context-conditioned distributions show low pairwise JS divergence on average, while target support remains high. The fraction satisfying

both reliability thresholds is highest for edit queries (95.0%) and lower for paraphrase (75.6%) and generation queries (73.1%).

Table 5: Reliability statistics of the knowledge-conditioned behavioral reference. Reliable rate denotes the percentage of queries satisfying $d_{\text{ctx}} \leq 0.1$ and $p_{\text{new}}^{\text{ctx}} \geq 0.2$.

Query type	**Pairwise JS** $\downarrow$	$p_{\text{new}}^{\text{ctx}} \uparrow$	**Reliable rate (%)**
Edit	0.040	0.818	95.0
Paraphrase	0.055	0.841	75.6
Generation	0.060	0.796	73.1

C Additional Experimental Results

C.1 Representation–Behavior Gap Analysis

We further analyze whether the benefit of BECAS varies with the representation–behavior mismatch of individual edits. Following Eq. 6.2, we compute G_e for each edit at the fixed pre-calibration operating point $l = 42$ and $\alpha = 1$. A larger G_e indicates that behavioral discrepancy is high relative to representation discrepancy.

For the $1{,}000$ CounterFact edits used in this analysis, we sort the edits by G_e and partition them into Low, Medium, and High groups. Table 6 reports the number of edits in each group. The corresponding editing results are reported in Table 2.

Table 6: Number of CounterFact edits in each representation–behavior gap group.

	Low	**Medium**	**High**
Cases	326	318	356

C.2 Evaluation on LLaMA-3-8B-Instruct

To examine whether the representation–behavior gap and the effectiveness of BECAS generalize beyond GPT2-XL, we further evaluate BECAS on LLaMA-3-8B-Instruct using CounterFact. We follow the same calibration–evaluation protocol and evaluation metrics as in the main experiments in Table 2. The underlying representation transport is instantiated using SAKE, while BECAS calibrates the intervention layer and steering strength according to the knowledge-conditioned behavioral reference. On LLaMA-3-8B-Instruct, BECAS reduces Behavior JS from 24.52 to 15.63

Table 7: Results on LLaMA-3-8B-Instruct with CounterFact.

Method	Eff.↑	Gen.↑	Loc.↑	B.JS↓	OS↓
SAKE	94.78	82.13	86.54	24.52	20.93
BECAS	93.11	88.76	89.73	15.63	10.34

and OS from 20.93 to 10.34, while improving generalization from 82.13% to 88.76% and locality from 86.54% to 89.73%. These results show that the benefit of behavioral calibration extends beyond GPT2-XL.