GRADIENTS FOR INTERVENTIONS AND ACTIVATIONS FOR DETECTION: TARGETED FEATURE LEARNING IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-internal features can be studied through both their ability to identify a specified concept and their causal effect when manipulated, e.g., through steering or weight editing. A prominent approach to feature learning is Sparse Autoencoders (SAEs), which learn broad feature dictionaries whose relation to particular concepts is typically identified post hoc. However, many interpretability questions are instead hypothesis-driven and concern a concept specified in advance. We study this setting as *targeted feature learning*, where a single feature is constructed for such a predefined concept. We present a controlled comparison across three model signals (activation values, activation gradients, and parameter gradients) and two estimators (contrastive mean and a learned one-dimensional encoder-decoder), yielding six targeted methods, with CAA and GRADIEND as existing instances and four new methods covering the remaining combinations. We compare these methods against pretrained SAEs across 15 tasks and three language models, evaluating both detection and causal intervention. Across models, the strongest detection performance is achieved by contrastive activation value methods, whereas the strongest intervention performance is achieved by gradient-based methods. Overall, our results show that targeted feature quality depends jointly on the model signal and estimator, with detection and intervention capturing complementary properties.

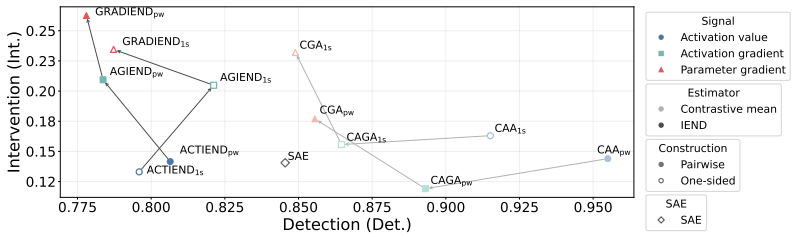


Figure 1: **Detection and intervention capture complementary properties of targeted features.** Across 15 tasks and three language models, activation value methods show stronger detection and most gradient-based methods stronger intervention overall. Per-model results are shown in Figure 3.

1 INTRODUCTION

Sparse Autoencoders (SAE; Bricken et al. (2023); Huben et al. (2024)) have become a prominent feature learning approach in mechanistic interpretability (Templeton et al., 2024), learning large dictionaries of activation features whose relation to particular concepts is typically identified post hoc (Arad et al., 2025). Many scientific questions are instead hypothesis-driven, targeting linguistic (Jing et al., 2025; Drechsel et al., 2026), semantic (Diera & Scherp, 2026), or demographic (Shan & Mueller, 2026) features, and testing whether they generalize and causally influence model behavior (Zou et al., 2023; Wu et al., 2025).

This motivates a complementary setting that we call **targeted learning**: learning a *single* feature for a concept specified in advance. Besides avoiding post-hoc feature selection, targeting a desired

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

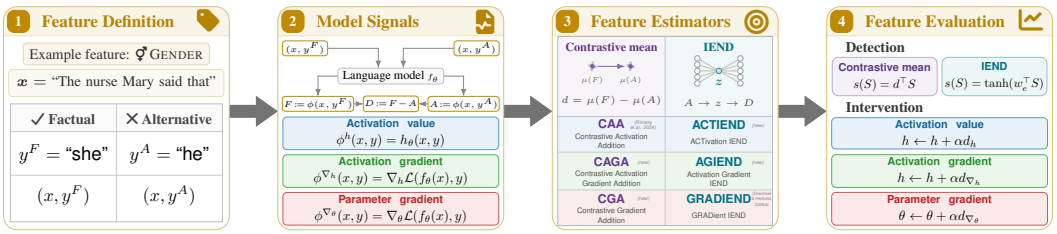


Figure 2: **Study overview.** From a predefined feature specification, we cross three model signals and two estimators, yielding six targeted feature learning methods evaluated on detection and intervention. SAEs provide an untargeted activation value reference closest to ACTIEND.

level of abstraction can address *feature splitting* and *feature absorption* in SAEs, which may divide broader concepts or leave systematic gaps in otherwise interpretable features (Chanin et al., 2025; Bussmann et al., 2025). A targeted feature should distinguish the specified concept from relevant alternatives and support interventions that causally change the corresponding model behavior.

Existing targeted methods differ along several dimensions. CAA (Rimsky et al., 2024) constructs feature directions from contrastive activation means, whereas GRADIEND (Drechsel & Herbold, 2026a) learns targeted features from contrastive parameter gradients. These representative approaches therefore differ simultaneously in the model *signal* from which the feature is constructed (e.g., activations or gradients), the *estimator* used to extract it (e.g., a contrastive mean or a learned model), and the *space* in which intervention is performed (e.g., activation or parameter space). A direct comparison cannot determine which of these choices accounts for differences in feature quality.

We disentangle these factors through a controlled 3×2 comparison (Figure 2). We consider *activation values*, *activation gradients*, and *parameter gradients* as signals, and pair each with either a *contrastive mean* estimator or the one-dimensional *Interpretable ENcoder Decoder (IEND)* estimator generalized from GRADIEND. We introduce activation gradients as a bridge between existing approaches by incorporating gradient information while retaining activation space intervention, allowing us to separate the effect of gradient information from parameter space intervention.

AxBench (Wu et al., 2025) provides the closest existing benchmark of feature representations through detection and steering, but its construction is not directly transferable to our gradient-based methods: it aggregates activations across the entire input sequence, while our targeted methods construct features at the feature-inducing target position. AxBench further relies on LLMs for feature data generation and steering evaluation. In contrast, our benchmark avoids this LLM dependence and further evaluates representations throughout model depth rather than at only two fixed layers, an important distinction given the substantial layerwise variation we observe (Appendix C.2).

To evaluate this target-localized setting broadly, we compare all six targeted approaches against SAEs as an untargeted reference across 15 diverse tasks (⚥GENDER, ⇄IOI, LANGID, ...) and three open-weight language models. Following AxBench, we evaluate two complementary dimensions: *detection*, measuring how well the learned feature distinguishes the specified concept, and *intervention*, measuring how effectively manipulating the feature causally changes model behavior.

Across models and tasks, we find a recurring trade-off between detection and intervention (Figure 1): activation value methods highlighted by CAA achieve the strongest detection, whereas gradient-based methods highlighted by GRADIEND achieve the strongest intervention. The intermediate methods help disentangle the effects of signal and estimator. Replacing the contrastive mean with IEND consistently reduces detection and, for gradient signals, increases intervention. Within a fixed estimator family, moving from activation values toward parameter gradients increases intervention. For contrastive mean methods, this comes with a loss in detection. Overall, feature quality depends jointly on signal and estimator.

Contributions. We systematically study targeted feature learning as a framework for hypothesis-driven feature analysis and provide a controlled comparison of model signal and feature estimator, introducing ACTIEND, AGIEND, CAGA, and CGA to complete the resulting 3×2 design. Across 15 tasks, three language models, and pretrained SAEs as an untargeted reference, we characterize how these choices differently affect detection and causal intervention.

2 Related work

Targeted concept representations and activation steering. A long line of work studies researcher-specified concepts through linear directions in model activations. Concept Activation Vectors (Kim et al., 2018) learn supervised concept directions, while representation engineering methods (Zou et al., 2023) and Activation Addition (Turner et al., 2025) use activation directions for analysis and steering. Contrastive Activation Addition (CAA) (Rimsky et al., 2024) constructs directions from paired contrastive examples, and closely related mean-difference directions have been used for targeted concept detection (Marks & Tegmark, 2024; Wu et al., 2025). More generally, probing establishes decodability without necessarily demonstrating a coherent or causally used feature (Hewitt & Liang, 2019; Belinkov, 2022). Park et al. (2024) connects linear representations defined by counterfactual pairs to both probing and steering, closely matching our use of controlled factual-alternative pairs. Other work optimizes supervised representation interventions or studies activation-steering geometry (Wu et al., 2024; Singh et al., 2024; Im & Li, 2026; Braun et al., 2025). Most closely, AxBench (Wu et al., 2025) evaluates both concept detection and steering across representation methods, including mean-difference directions and SAEs. Our study instead systematically varies the signal space and estimator used for feature construction, extending the comparison from activation values to activation gradients and parameter gradients.

Sparse feature learning. Sparse Autoencoder (SAE; Bricken et al. (2023); Huben et al. (2024); Templeton et al. (2024)) learn large activation feature dictionaries without specifying the concept of interest in advance and thus provide an untargeted reference for our setting. Identifying a concept post hoc can be complicated by feature splitting and absorption, variation across SAE trainings, and concepts distributed across multiple latents (Chanin et al., 2025; Leask et al., 2025; Paulo & Belrose, 2026; Bhalla et al., 2026), making intervention dependent on latent selection and combination (Chalnev et al., 2024; Arad et al., 2025; Soo et al., 2025; Jørgensen & Hansen, 2026). Recent work brings concept supervision into SAE training, moving closer to the targeted setting considered here. Sparse Conditioned Autoencoders (SCAR; Härle et al. (2024)) supervises a designated latent, while Guided SAEs (Härle et al., 2025) encourage specified concepts to localize in the sparse representation. Both retain a larger SAE dictionary rather than constructing only the specified feature.

Gradient-based feature construction and intervention. While SAEs learn dictionaries of activation features, GRADIEND (Drechsel & Herbold, 2026a) uses a one-dimensional encoder-decoder to learn a single targeted feature from factual-alternative parameter gradients, and uses its decoder direction for persistent parameter intervention. Parameter space directions are also used in task arithmetic (Ilharco et al., 2023), while activation gradients have previously supported inference-time control (Dathathri et al., 2020; Oozeer et al., 2025; Sharma & Trivedi, 2026). In contrast, our activation gradient methods treat such gradients as the signal from which a reusable targeted feature is learned across factual-alternative contrasts. Notably, Gradient SAEs (Shu et al., 2025), despite the name, learn features from activations and use output gradients only to rank their estimated influence.

3 Targeted feature learning

3.1 From feature discovery to targeted feature learning

We study one-dimensional *targeted features* for concepts specified before feature construction. Each feature comprises a scalar score for *detection* and an associated model-internal direction for *intervention*, following the one-dimensional abstraction common to concept-vector and representation-steering methods (Kim et al., 2018; Zou et al., 2023). *Detection* asks whether the score distinguishes the specified feature realizations, whereas *intervention* asks whether applying the intervention vector causally changes model behavior in the corresponding feature direction (Wu et al., 2025).

We call feature learning *targeted* when the feature specification is used during feature construction, rather than only to identify or interpret a representation post hoc. SAEs (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2024) provide an untargeted reference, since their features are learned independently of the target concept, after which examples expressing that feature can be used to identify the corresponding latent. Targeted methods instead use these examples during construction.

Table 1: Targeted feature learning methods across three model-derived *signals* and two *estimators*.

Signal	Contrastive mean	Interpretable ENcoder Decoder (IEND)
Activation value	Contrastive Activation Addition (CAA) (Rimsky et al., 2024)	ACTivation IEND (ACTIEND)[†] *(new)*
Activation gradient	Contrastive Activation Gradient Addition (CAGA) *(new)*	Activation Gradient IEND (AGIEND) *(new)*
Parameter gradient	Contrastive Gradient Addition (CGA) *(new)*	GRADient IEND (GRADIEND)[*](Drechsel & Herbold, 2026a)

[†] Pretrained SAEs provide an untargeted activation value reference most directly comparable to ACTIEND.
[*] GRADIEND originally stood for *GRADIent ENcoder Decoder*. With IEND as the generalized estimator family, we use *GRADient IEND*.

Here, supervision consists of matched factual and alternative targets that instantiate different feature realizations while sharing the same context. Section 3.2 formalizes these contrasts.

Given the same feature specification, targeted methods can differ in *what* model-derived information they use and *how* they construct a one-dimensional feature from it. Contrastive Activation Addition (CAA; Rimsky et al. (2024)) and GRADIEND (Drechsel & Herbold, 2026a) provide two useful anchors for these choices because both construct a single concept-specific feature from contrastive supervision, but make fundamentally different choices for both signal and estimator. CAA constructs a contrastive mean direction from activation values, whereas GRADIEND learns a one-dimensional encoder-decoder with a single scalar feature neuron from parameter gradient signals. A direct comparison between them therefore changes both the *signal* from which the feature is learned and the *estimator* used to learn it. We disentangle them by crossing activation values, activation gradients, and parameter gradients with either a contrastive mean estimator or the GRADIEND-generalized one-dimensional Interpretable ENcoder Decoder (IEND). This yields the six targeted methods summarized in Table 1.

3.2 Specifying features through contrasts

We specify a feature through different realizations of that feature. For a feature with K classes, let $\mathcal{C} \coloneqq \{c_1, \ldots, c_K\}$ and $\mathcal{D}_k$ denote natural examples whose factual target realizes c_k. For each example x_i, we denote this factual target by y_i^F and construct an alternative target y_i^A for the same input such that the feature realization changes while the surrounding context is held fixed. We refer to the resulting factual-alternative comparison (y_i^F, y_i^A) as a *contrast*. By controlling what differs within this contrast, we specify the feature to be learned. We consider two contrast constructions:

$$\begin{aligned} &\textbf{Pairwise } (c_k, c_m): && x_i \in \mathcal{D}_k \cup \mathcal{D}_m, && y_i^A \text{ realizes the other of } c_k, c_m, \\ &\textbf{One-sided } (c_k): && x_i \in \mathcal{D}_k, && y_i^A \text{ realizes a non-target alternative.} \end{aligned} \tag{1}$$

Pairwise learning is symmetric: factual examples are drawn from both classes, with the factual and alternative roles reversed across them. It yields a polar axis between c_k and c_m, oriented from c_k (lower scores) to c_m (higher scores), with intermediate scores indicating weaker alignment with either class. This construction is used by CAA and GRADIEND.

One-sided learning instead uses natural examples only from the target class c_k and learns its presence relative to valid non-target alternatives. We orient the feature such that higher scores indicate stronger expression of c_k. For categorical features, alternatives are drawn from $\mathcal{C} \setminus \{c_k\}$. This also covers binary concepts for which natural data are available only for the positive class: the alternative represents absence of the feature.

For $K = 2$, pairwise learning yields one symmetric polar feature, while one-sided learning yields two independently learned target-presence features. For $K \geq 2$, the constructions yield $\binom{K}{2}$ pairwise and K one-sided features, respectively. Thus, pairwise learning targets particular class-to-class distinctions, while one-sided learning provides a more compact class-specific representation.

Let ϕ map an input-target pair to the model-derived signal used for feature learning. Each factual-alternative contrast induces $F_i = \phi(x_i, y_i^F)$, $A_i = \phi(x_i, y_i^A)$, and $D_i = F_i - A_i$, where F_i and A_i are the signals induced by the factual and alternative targets, respectively. We next instantiate ϕ in three signal spaces and use these signals with either feature estimator.

3.3 THREE SIGNAL SPACES

We instantiate each contrast in three model signal spaces. Let f_θ denote the model, Ω the selected internal components (e.g., residual stream activations or transformer block parameters), h_Ω their activations, and θ_Ω the corresponding parameters. We define

$$\phi^{h}(x, y) \coloneqq h_\Omega(x, y), \qquad \phi^{\nabla_h}(x, y) \coloneqq \nabla_{h_\Omega} \mathcal{L}(f_\theta(x), y), \qquad \phi^{\nabla_\theta}(x, y) \coloneqq \nabla_{\theta_\Omega} \mathcal{L}(f_\theta(x), y).$$

Activation values provide the internal representation itself, activation gradients characterize how the objective locally changes with that representation, and parameter gradients how it locally changes with the selected parameters. Model scopes, readouts, and objectives are specified in Section 4.

The signal spaces also differ in when target information becomes available. For gradient-based signals, the candidate target enters through the objective, so gradients can be extracted from the model state preceding the target. For activation values, factual and alternative targets induce the same pre-target activation. Targeted learning methods based on activation values therefore require the candidate target to be filled in before extracting the learning signal (i.e., "he" and "she" must be part of the input in Figure 2). Intervention, however, must act before the corresponding prediction is made. Pretrained SAEs are not subject to this restriction because their features are learned independently of the factual-alternative contrast. Appendix C.3 compares pre-target and filled-target activations and finds no consistent intervention advantage for pre-target activations.

3.4 TWO TARGETED FEATURE ESTIMATORS

We compare the two estimator families underlying the established targeted methods CAA and GRADIEND: a contrastive mean and a learned one-dimensional encoder-decoder. Both yield scalar features for detection and intervention across all three signal spaces, while differing in whether the feature is estimated directly from contrastive differences or learned with a reconstruction objective.

Contrastive mean. For each feature, we fix a positive and a negative semantic pole. For a pairwise feature (c_k, c_m), c_k is the negative and c_m the positive pole. For a one-sided feature, the target class is the positive pole. Recall that $D_i = F_i - A_i$. For pairwise features, factual and alternative targets exchange roles across examples from the two classes, so the same semantic class contrast appears in opposite signs in D_i. We therefore introduce $o_i \in \{-1, +1\}$ and set $o_i = +1$ when y_i^F realizes the positive pole and $o_i = -1$ otherwise, such that $o_i D_i$ always points from the negative to the positive realization. For one-sided features, the factual target is always positive, so $o_i = +1$. The contrastive mean direction is $d_{\text{mean}} \coloneqq \frac{1}{N} \sum_{i=1}^{N} o_i D_i$. Under a balanced pairwise construction, this is equivalent to a difference-of-means estimator. Applied to activation values, activation gradients, and parameter gradients, it yields CAA (Rimsky et al., 2024), CAGA, and CGA, respectively.

Interpretable ENcoder Decoder (IEND). We generalize the encoder-decoder construction introduced by GRADIEND (Drechsel & Herbold, 2026a) into an estimator that can be applied across different signal spaces, which we call IEND. Rather than averaging the transitions into a single vector, IEND learns to reconstruct them from corresponding source signals through a single scalar bottleneck. This encoder-decoder structure is closely related to an individual SAE feature (Bricken et al., 2023; Huben et al., 2024): both associate a scalar encoder value with a decoder vector. While an SAE learns a large dictionary through untargeted activation reconstruction, IEND uses the predefined feature specification to learn only a single scalar feature and its corresponding decoder. It therefore acts as a targeted analogue of learning an individual SAE-like feature rather than a dictionary from which the desired feature must later be identified.

Let (S, T) denote the signal provided to the encoder and the transition reconstructed by the decoder. In pairwise learning, examples from both classes already expose both feature poles as sources: across contrasts, A_i spans both classes and is paired with D_i. In one-sided learning, factual signals all belong to the target side, while alternative signals represent the non-target side. Using only (A_i, D_i) would therefore expose the encoder only to non-target sources. We instead use both realizations as sources: (A_i, D_i) and $(F_i, -D_i)$. Let $\mathcal{P}$ denote the resulting set of source-target pairs. The encoder maps each source signal to a scalar feature value, while the decoder reconstructs the corresponding transition:

$$z = \text{enc}(S) \coloneqq \tanh(w_e^\top S + b_e), \qquad \text{dec}(z) \coloneqq b_d + w_d z.$$

Table 2: Summary of the 15 feature learning tasks. pw denotes pairwise and $1s$ one-sided tasks.

Family	Eval.	Tasks (classes)
Semantic	pw/$1s$	GENDER (male/female); EMOTION (positive/negative); RACE (Asian/Black/White); RELIGION (Christian/Muslim/Jewish)
Factual	pw/$1s$	RAVELCONT (Asia/Europe/Africa); RAVELCNTRY (United States/China/Russia); RAVELLANG (English/Portuguese/Spanish)
Language	pw/$1s$	LANGID (English/French/German)
Grammatical	pw/$1s$	PRONNUM (singular/plural); PRONPERS (first/second/third)
Algorithmic	$1s$	IOI (IO/subject); KEYVALUE (value/other value); INDUCTION (match/distractor); REPETITION (yes/no); FUNCCOMP (result/distractor)

Encoder and decoder are jointly optimized to minimize $\mathcal{L} = \frac{1}{|\mathcal{P}|} \sum_{(S,T)\in\mathcal{P}} \|T - \mathrm{dec}(\mathrm{enc}(S))\|_2^2$. Applied to activation values, activation gradients, and parameter gradients, this yields ACTIEND, AGIEND, and GRADIEND, respectively.

3.5 FEATURE DETECTION AND FEATURE INTERVENTION

We use each learned feature for both *detection* and *intervention*. For contrastive mean methods, the feature score is $s_{\mathrm{mean}}(S) := \frac{d_{\mathrm{mean}}^\top S}{\|d_{\mathrm{mean}}\|_2 \, \|S\|_2}$. For IEND, it is the encoder output $s_{\mathrm{IEND}}(S) := \mathrm{enc}(S)$. For intervention, we orient the learned transition toward a selected class c_i. For contrastive mean methods, this means orienting d_{mean} toward c_i; for IEND, we use the decoder output at the corresponding feature value z_{c_i}. Let d_{c_i} denote the resulting class-oriented intervention direction. We intervene with $h_\Omega \leftarrow h_\Omega + \alpha d_{c_i}$ in activation space and $\theta_\Omega \leftarrow \theta_\Omega + \alpha d_{c_i}$ in parameter space, with intervention strength $\alpha \in \mathbb{R}$. Positive α moves toward c_i, while negative α moves in the opposite direction.

Sparse autoencoder reference. For SAE, we identify the requested feature post hoc by ranking latents by their ability to distinguish it. The selected latent activation provides the detection score and its decoder vector the activation-space intervention direction. We treat each selected latent as a one-sided feature, steering toward or away from its decoder direction.

4 EXPERIMENTAL SETUP

Models and tasks. We evaluate GPT-2 (Radford et al., 2019), Pythia-70M (Biderman et al., 2023), and Llama-3.1-8B (Grattafiori et al., 2024) on 15 tasks spanning semantic, grammatical, factual, language, and algorithmic features. The benchmark includes both pairwise and one-sided feature specifications (Section 3.2). For the SAE baseline, we use pretrained SAEs from SAELens (Bloom et al., 2024). The models were selected based on the availability of pretrained SAEs and to cover pretraining from different model providers at different model sizes. Table 2 summarizes the tasks. Details including dataset constructions are reported in Appendix A.

Protocol and model representations. All methods use the same train-validation-test splits. Training data constructs the features, validation data selects method-specific choices, and test data is used only for final evaluation. Validation selection includes IEND checkpoints, SAE latents, detection thresholds, and intervention strengths. Activation value methods use residual stream activations after each selected transformer block, averaged over the positions corresponding to the target tokens. Gradient methods condition on the first target token: activation gradients are taken with respect to the residual stream at its prediction position and parameter gradients with respect to transformer block parameters. We exclude embeddings and the final output head. Details are in Appendix B.

Detection evaluation. Detection measures how well a feature score separates its target class from neutral examples and, where applicable, other classes of the same feature, which we call *rivals*. For target class c_k, let AUC_n denote target-versus-neutral AUROC and AUC_o the lowest target-versus-rival AUROC. At a validation-selected threshold, Spec_n is specificity on neutral examples and Excl the lowest specificity across rivals. We define

$$\mathrm{Det.} = \begin{cases} \min\{\mathrm{AUC}_n, \mathrm{AUC}_o, \mathrm{Spec}_n, \mathrm{Excl}\} & \text{with rivals} \\ \min\{\mathrm{AUC}_n, \mathrm{Spec}_n\} & \text{without rivals.} \end{cases}$$

Taking the minimum requires a feature to perform well on all applicable detection criteria rather than allowing strong performance on one criterion to compensate for failure on another while still yielding a single scalar score.

Intervention evaluation. Intervention measures how manipulating the feature changes the model's probability of the target behavior For a feature oriented toward target class t, let I^+ denote a strengthening intervention toward t, I^- a weakening intervention in the opposite direction, and I_0 the unmodified model. Let $\bar{P}_I(t)$ denote the mean next-token probability assigned to t across the corresponding held-out examples under intervention I. For independently selected strengthening and weakening interventions,

$$\Delta P^+ \coloneqq \bar{P}_{I^+}(t) - \bar{P}_{I_0}(t), \qquad \Delta P^- \coloneqq \bar{P}_{I_0}(t) - \bar{P}_{I^-}(t), \qquad \text{Int.} \coloneqq (\Delta P^+ + \Delta P^-)/2.$$

For pairwise features, strengthening is evaluated on rival examples and weakening on target examples. For one-sided features, both use target examples. To prevent large effects caused by broad model degradation, strengthening and weakening magnitudes are selected independently on validation data subject to a language modeling constraint. (Meng et al., 2022). With $\mathrm{LMS} \coloneqq 1/(1+\overline{\mathrm{NLL}})$, where $\overline{\mathrm{NLL}}$ is the mean token-level negative log-likelihood, we require $\mathrm{LMS}(I) \geq 0.99\,\mathrm{LMS}(I_0)$ and independently select strengthening and weakening interventions by grid search over approximately log-spaced magnitudes, maximizing the validation effect.

Representation selection. CAA and pretrained SAEs are conventionally evaluated as layer-local activation features (Lawson et al., 2025; Lieberum et al., 2024), and their performance can depend strongly on layer choice (Rimsky et al., 2024; Al Ghussin et al., 2026). An individual-layer scope uses the post-block residual stream representation after transformer block ℓ for activation-based methods, and the corresponding block parameters for parameter-gradient methods. For SAE and contrastive mean methods, we evaluate all single layer scopes and an all-layer scope, the latter enabling direct comparison with IEND, which learns jointly across selected components. Validation detection selects among these candidates, and the selected scope is fixed for both test detection and intervention. For IEND, evaluating every layer separately would require training a distinct encoder-decoder for each scope, so we use fixed joint scopes: all post-block residual stream activations for ACTIEND and AGIEND, and the joint parameter gradient scope for GRADIEND.

Aggregation. For each method, model, and task, detection and intervention scores are first computed for each evaluated class or contrast member and then averaged. Task scores are further averaged within each model, and cross-model summaries average the resulting model means.

5 Feature detection and intervention performance

Figures 1 and 3 show the model-averaged and corresponding per-model results. Activation value methods, particularly CAA, achieve the strongest detection, while gradient-based methods, particularly GRADIEND, achieve the strongest intervention overall. Thus, detection and intervention capture complementary properties. The SAE reference is comparatively modest on both measures and does not match the strongest targeted methods on either. Detailed results are in Appendix C.

Signal and estimator. Within the contrastive mean family, CGA yields substantially stronger intervention than CAA at the cost of detection, while CAGA shows a less consistent intermediate pattern: it intervenes better than CAA for GPT-2, but less effectively for Pythia-70M and Llama-3.1-8B. For gradient signals, replacing the contrastive mean with IEND further typically increases intervention while often reducing detection. For activation values, however, replacing the contrastive mean with IEND typically reduces both detection and intervention, moving ACTIEND below CAA on both measures. ACTIEND and the pretrained SAE reference, the two learned activation value approaches, occupy a similar overall region.

Feature construction. The aggregate figure is not directly comparable along this axis because one-sided methods include additional tasks. On matched tasks, one-sided construction generally matches pairwise performance and often yields slightly higher detection, with similar intervention for most methods (Appendix C.5). One-sided construction additionally requires only K rather than $\binom{K}{2}$ feature models and supports features without a natural opposing class.

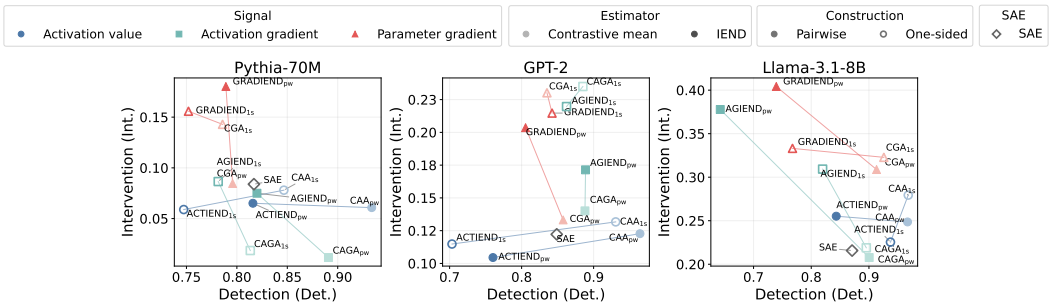
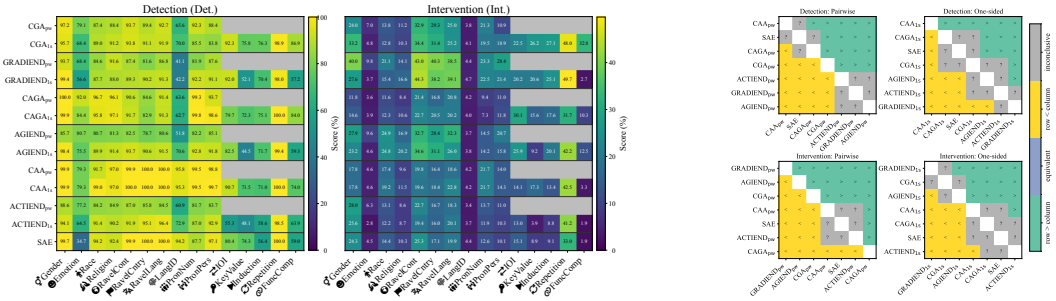


Figure 3: Task-averaged detection and intervention for each method. Lines connect the two estimators using the same signal and construction. The model-averaged aggregate is shown in Figure 1.



(a) Model-averaged Det. and Int. scores for each task and method.

(b) Bayesian method comparison.

Figure 4: Overview of task-level performance and Bayesian method comparisons.

Task-level variation. Figure 4a shows substantial variation across tasks. LANGID and EMOTION are the most difficult for both detection and intervention. For EMOTION, the SAEs fails to recover a feature latent, whereas the targeted methods retain non-trivial detection. Most remaining pairwise capable tasks are detected reliably across methods, whereas several one-sided tasks are substantially harder, with some methods approaching chance-level detection (e.g., KEYVALUE). Intervention varies more strongly across tasks and methods than detection: strong detection does not consistently imply strong intervention. The RAVEL tasks, GENDER, and REPETITION show some of the largest intervention effects, particularly for gradient-based methods.

Bayesian method comparison. To complement the mean scores with a comparison that is less dominated by the magnitude of isolated method-task differences, we also perform a Bayesian pairwise analysis over the matched model-task observations using `autorank` (Herbold, 2020). Figure 4b supports our main conclusion: activation value methods occupy the strongest positions for detection, while GRADIEND is strongest for pairwise intervention, followed by AGIEND. Under one-sided construction, the two remain statistically unresolved. Details are given in Appendix B.8 and C.1.

Representation scope. We next compare features represented at a single transformer layer with those using the corresponding joint all-layer representation. For SAE and contrastive mean methods, validation detection selects a single layer over the all-layer representation in 93.0% of class/contrast decisions and the selected layer also achieves higher held-out detection in 80.4% of comparisons (Appendix C.2). Intervention differences are smaller and less systematic. Moreover, repeating the main detection-intervention comparison from Figure 3 with SAE and contrastive mean methods restricted to their all-layer representations preserves the main signal-level pattern: activation value methods remain strongest for detection, while gradient-based methods remain strongest for intervention (Figure 8). Thus, our key observation is not driven by the representation-scope difference of contrastive mean methods relative to IEND.

6 Discussion and Limitations

Detection versus intervention. Our results qualify the widespread use of activation value features for both feature identification and model steering. The strongest detection methods use activation values, whereas the strongest intervention methods use gradient-based signals. This finding is especially notable given the prominence of activation-based steering approaches in prior work. CAA was introduced as a targeted activation-steering method (Rimsky et al., 2024), while SAEs are widely used for feature discovery and have also been applied to causal intervention (Bricken et al., 2023; Huben et al., 2024; Templeton et al., 2024). Yet in our evaluation, CAA is particularly strong for detection, whereas AGIEND and GRADIEND are particularly strong for intervention. AGIEND further shows that this advantage does not require parameter space intervention.

Positional asymmetry. An important consideration concerns where these signals become observable. In our tasks, activation value contrasts are measured at the feature-related target representation: factual and alternative examples become distinguishable only once the corresponding target token (e.g., a gender pronoun for ⚥GENDER) enters the residual stream. Intervention, however, must influence that target before or while it is generated. Gradient signals do not share this restriction, since the candidate target enters through the CLM objective and its gradient signal can be derived at the positions predicting it. This positional asymmetry is a consequence of our targeted feature learning setup: when factual and alternative examples share the same prefix, their pre-target activations are identical, so no targeted activation contrast can be learned. Untargeted SAE features do not have this restriction because they are learned independently of the factual-alternative contrast. Nevertheless, selecting SAE features from pre-target activations does not consistently improve intervention (Appendix C.3). Thus positional alignment may contribute to, but does not fully explain, the intervention advantage of gradient signals.

Decoupling detection and intervention. The separation between detection and intervention suggests that identifying and manipulating a feature need not use the same signal or representation scope. Activation values directly reflect whether a feature is present in the model state, while gradients encode how the objective changes with respect to that state or its parameters. Our representation-scope analysis (Figure 7) points in the same direction: detection benefits strongly and consistently from localizing a feature to an informative single layer, whereas intervention is considerably less sensitive to the same localization. A natural extension of IEND is therefore to decouple these roles, for example using localized activations as encoder inputs and contrastive gradients as decoder targets, potentially with different input and output scopes. Such a hybrid could combine strong activation-based readout with gradient information useful for intervention. Whether these advantages can be combined in a single feature learner, or whether their trade-off between detection and intervention is more fundamental, remains an open empirical question.

Limitations. Our evaluation covers three language models and 15 tasks, and generalization to other architectures, scales, and feature types remains to be established. The target-localized formulation enables controlled comparison across methods but differs from broader benchmarks such as AxBench (Wu et al., 2025), preventing a direct matched comparison. Finally, intervention is evaluated at the specified target token and does not capture effects on longer generations.

7 Conclusion

We presented a controlled comparison of targeted feature learning methods that separates three design choices: the model signal, the feature estimator, and the feature specification. Across three language models and 15 tasks, detection and intervention emerge as complementary measures of feature quality. Contrastive activation value features provide particularly strong detection, while gradient-based methods achieve the strongest causal intervention, especially as learned IEND features. Feature construction and representation scope further affect this trade-off: one-sided formulations broaden the applicability of targeted feature learning, while layer localization benefits detection considerably more consistently than intervention. Together, these findings suggest that identifying and causally manipulating a feature need not rely on the same signal or representation scope. They motivate targeted feature learning methods that explicitly combine these complementary strengths, including hybrid and layer-selective IEND formulations.

AI USE STATEMENT

Generative AI tools were used to discuss and refine aspects of the research methodology and experimental design, assist with the implementation of several methods following author-specified designs, and support interpretation of experimental results. Additionally, generative AI tools were used for literature discovery, generation of initial drafts of parts of the manuscript, and editing of the text. All AI-assisted code was reviewed and tested by the authors, suggested references were checked against the original sources, and AI-assisted text was reviewed and substantially revised. Generative AI tools were not used to generate synthetic datasets. The authors take responsibility for the final content of this work, including all claims, results, code, and text produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We release the full study code (Appendix B.1) and all redistributable task datasets (Tables 3 and 5), including the extended GRADIEND package code, experiment configurations, and analysis scripts. Synthetic datasets can be regenerated deterministically from fixed seeds, while tasks based on third-party data are reconstructed from their public sources using the provided scripts. The repository README documents installation, dataset generation, and the commands for reproducing the main experiments. Appendices A and B provide further details on the tasks, models, signals, training, evaluation, aggregation, layer selection, and statistical analysis. Representative compute costs for two models on one task are reported in Appendix B.3. All training and evaluation runs use fixed seeds, with model and method selection performed on validation data and test data used only for final evaluation.

REFERENCES

Yusser Al Ghussin, Daniil Gurgurov, Tanja Baeumel, Josef van Genabith, Patrick Schramowski, and Simon Ostermann. Multilingual steering by design: Multilingual sparse autoencoders and principled layer selection. In Kai-Wei Chang, Ninareh Mehrabi, Satyapriya Krishna, Anubrata Das, Jwala Dhamala, Yang Trista Cao, Tharindu Kumarage, Anil Ramakrishna, Christos Christodoulopoulos, Yixin Wan, Aram Galystan, Anoop Kumar, and Rahul Gupta (eds.), *Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026)*, pp. 364–401, San Diego, California, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-418-7. doi: 10.18653/v1/2026.trustnlp-main.24. URL https://aclanthology.org/2026.trustnlp-main.24/.

Dana Arad, Aaron Mueller, and Yonatan Belinkov. SAEs are good for steering – if you select the right features. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10241–10259, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.519. URL https://aclanthology.org/2025.emnlp-main.519/.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7/.

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. Do sparse autoencoders capture concept manifolds?, 2026. URL https://arxiv.org/abs/2604.28119.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. URL https://proceedings.mlr.press/v202/biderman23a/biderman23a.pdf.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. Saelens. https://github.com/decoderesearch/SAELens, 2024.

Joschka Braun, Carsten Eickhoff, David Krueger, Seyed Ali Bahrainian, and Dmitrii Krasheninnikov. Understanding (un)reliability of steering vectors in language models. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=JZiKuvIK1t.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Bart Bussmann, Noa Nabeshima, Adam Karvonen, and Neel Nanda. Learning multi-level features with matryoshka sparse autoencoders. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6077–6101. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/bussmann25a.html.

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features, 2024. URL https://arxiv.org/abs/2411.02193.

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 82318–82355. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/764ff7477b8e24dbe01531f6791e8bdf-Paper-Conference.pdf.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *International Conference on Learning Representations*, 2020. URL https://mlanthology.org/iclr/2020/dathathri2020iclr-plug/.

Andor Diera and Ansgar Scherp. Do language models encode semantic relations? probing and sparse feature analysis. In Stelios Piperidis, Núria Bel, Henk van den Heuvel, Nancy Ide, Simon Krek, and Antonio Toral (eds.), *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pp. 2115–2126, Palma de Mallorca, Spain, May 2026. ELRA Language Resource Association. doi: 10.63317/52znrwwkdog7. URL https://aclanthology.org/2026.lrec-1.166/.

Jonathan Drechsel and Steffen Herbold. GRADIEND: Feature learning within neural networks exemplified through biases. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=1vBNAnAgCD.

Jonathan Drechsel and Steffen Herbold. The GRADIEND python package: An end-to-end system for gradient-based feature learning, 2026b. URL https://arxiv.org/abs/2602.23993.

Jonathan Drechsel, Erisa Bytyqi, and Steffen Herbold. Understanding or memorizing? a case study of German definite articles in language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9626–9652, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.436. URL https://aclanthology.org/2026.acl-long.436/.

Jack Gallifant, Shan Chen, Kuleen Sasse, Hugo Aerts, Thomas Hartvigsen, and Danielle Bitterman. Sparse autoencoder features for classifications and transferability. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29939–29963, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1521. URL https://aclanthology.org/2025.emnlp-main.1521/.

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

Ruben Härle, Felix Friedrich, Manuel Brack, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. SCAR: Sparse conditioned autoencoders for concept detection and steering in LLMs. In *Workshop on Socially Responsible Language Modelling Research*, 2024. URL https://openreview.net/forum?id=o6pgbB1ijo.

Ruben Härle, Felix Friedrich, Manuel Brack, Björn Deiseroth, Stephan Waeldchen, Patrick Schramowski, and Kristian Kersting. Measuring and guiding monosemanticity. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=REHjkmWdQL.

Steffen Herbold. Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48):2173, 2020. doi: 10.21105/joss.02173. URL https://doi.org/10.21105/joss.02173.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *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. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Jing Huang, Zhengxuan Wu, Christopher Potts, Mor Geva, and Atticus Geiger. RAVEL: Evaluating interpretability methods on disentangling language model representations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8669–8687, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.470. URL https://aclanthology.org/2024.acl-long.470/.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing Models with Task Arithmetic. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/ilharco2023iclr-editing/.

Shawn Im and Sharon Li. A unified understanding and evaluation of steering methods, 2026. URL https://arxiv.org/abs/2502.02716.

Yi Jing, Zijun Yao, Hongzhu Guo, Lingxu Ran, Xiaozhi Wang, Lei Hou, and Juanzi Li. LinguaLens: Towards interpreting linguistic mechanisms of large language models via sparse autoencoder. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 28232–28251, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1433. URL https://aclanthology.org/2025.emnlp-main.1433/.

Mikkel Godsk Jørgensen and Lars Kai Hansen. Steering llms? actually, sparse autoencoders can outperform simple baselines, 2026. URL https://arxiv.org/abs/2605.31183.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=qrU3yNfX0d.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2018. URL https://arxiv.org/abs/1711.11279.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H196sainb.

Tim Lawson, Lucy Farnik, Conor Houghton, and Laurence Aitchison. Residual stream analysis with multi-layer SAEs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=XAjfjizaKs.

Patrick Leask, Bart Bussmann, Michael T Pearce, Joseph Isaac Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9ca9eHNrdH.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.19. URL https://aclanthology.org/2024.blackboxnlp-1.19/.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36, 2022. URL https://arxiv.org/abs/2202.05262. arXiv:2202.05262.

Aaron Mueller, Atticus Geiger, Sarah Wiegreffe, Dana Arad, Iván Arcuschin, Adam Belfki, Yik Siu Chan, Jaden Fried Fiotto-Kaufman, Tal Haklay, Michael Hanna, Jing Huang, Rohan Gupta, Yaniv Nikankin, Hadas Orgad, Nikhil Prakash, Anja Reusch, Aruna Sankaranarayanan, Shun Shao, Alessandro Stolfo, Martin Tutek, Amir Zur, David Bau, and Yonatan Belinkov. MIB: A mechanistic interpretability benchmark. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=sSrOwve6vb.

Narmeen Fatimah Oozeer, Luke Marks, Fazl Barez, and Amir Abdullah. Beyond linear steering: Unified multi-attribute control for language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 23513–23557, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1278. URL https://aclanthology.org/2025.findings-emnlp.1278/.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EjInprGpk9.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

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

Zhengyang Shan and Aaron Mueller. Measuring mechanistic independence: Can bias be removed without erasing demographics? In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4241–4265, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.199. URL https://aclanthology.org/2026.eacl-long.199/.

Kartik Sharma and Rakshit Trivedi. Cold-steer: Steering large language models via in-context one-step learning dynamics. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 27736–27756, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/2ee1c87245956e3eaa71aaba5f5753eb-Paper-Conference.pdf.

Dong Shu, Xuansheng Wu, Haiyan Zhao, Mengnan Du, and Ninghao Liu. Beyond input activations: Identifying influential latents by gradient sparse autoencoders. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1673–1682, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.87. URL https://aclanthology.org/2025.emnlp-main.87/.

Shashwat Singh, Shauli Ravfogel, Jonathan Herzig, Roee Aharoni, Ryan Cotterell, and Ponnurangam Kumaraguru. Representation surgery: Theory and practice of affine steering. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 45663–45680. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/singh24d.html.

Samuel Soo, Wesley Teng, Chandrasekaran Balaganesh, Tan Guoxian, and Ming YAN. Interpretable steering of large language models with feature guided activation additions. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=swRxS7s4rB.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 2214–2218, Istanbul, Turkey, May 2012.

European Language Resources Association (ELRA). URL https://aclanthology.org/L12-1246/.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2025. URL https://openreview.net/forum?id=2XBPdPIcFK.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. URL https://arxiv.org/abs/1706.03762.

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

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=K2CckZjNy0.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1628–1639, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.148. URL https://aclanthology.org/2020.acl-main.148/.

Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023. doi: 10.48550/ARXIV.2310.01405. URL https://doi.org/10.48550/arXiv.2310.01405.

A Dataset Documentation

Table 3 provides an overview of the 15 evaluated tasks, including their feature classes, supported feature constructions, provenance, and available examples per class. Table 4 gives one representative example per task. Reported dataset sizes refer to the full available datasets and are given per class. The numbers of examples used for training, validation, detection evaluation, and intervention evaluation in this study are reported separately in Appendix B.5.

We distinguish three forms of dataset provenance. *Reused* tasks use existing processed datasets, *adapted* tasks transform existing benchmark or corpus data for our targeted feature learning setting, and *generated* tasks use controlled generators developed for this study. We release the newly generated datasets and their generators together with the study.

Prediction contexts and alternatives. Each task provides a context containing a `[MASK]` marker, a factual target, and its feature class. We retain `[MASK]` in Table 4 to make the target position and surrounding context explicit, even though this study uses decoder-only models that rely only on the left context up to the `[MASK]`.

Splits. For datasets whose splits are constructed as part of this study, we use an 80/10/10 train, validation, and test split, with the corresponding underlying entities kept disjoint where applicable. For reused datasets with predefined splits, such as ⇄IOI, we retain the original split.

Table 3: Complete task inventory. Sizes are effective labeled train/validation/test rows after task-level class mapping and before counterfactual expansion. P = both poles occur naturally and both pairwise and one-sided regimes are evaluated. O = only the first listed class is factual and the second is its constructed counterpart. We do not redistribute LANGID due to licensing constraints and instead provide a script for reconstructing it from the original sources.

Task	Feature classes	Reg.	Train/val/test	Source and Hugging Face ID
GENDER	male (M), female (F)	P	73,860/2,110/8,440	`aieng-lab/genter-ajibawa-name-filled` (Drechsel & Herbold, 2026a) subset `n1`
EMOTION	positive, negative	P	600/200/200	`aieng-lab/en-sentiment-nrc` (Drechsel & Herbold, 2026b) vocabulary-held-out split
RACE	Asian, Black, White	P	33,602/9,603/4,799	`aieng-lab/gradiend_race_data` (Drechsel & Herbold, 2026a)
RELIGION	Christian, Muslim, Jewish	P	20,048/5,730/2,863	`aieng-lab/gradiend_religion_data` (Drechsel & Herbold, 2026a)
RAVELCONT	Asia, Europe, Africa	P	1,344/168/168	Source: `hij/ravel` (Huang et al., 2024); Processed `anonymous` (new)
RAVELCNTRY	United States, China, Russia	P	408/51/51	Source: `hij/ravel` (Huang et al., 2024); Processed `anonymous` (new)
RAVELLANG	English, Portuguese, Spanish	P	528/66/66	Source: `hij/ravel` (Huang et al., 2024); Processed `anonymous` (new)
LANGID	English (en), French (fr), German (de)	P	24,000/3,000/3,000	Source: `Helsinki-NLP/opus-100` and `https://github.com/facebookresearch/MUSE`; Processed: `anonymous` (new)
PRONNUM	singular, plural	P	32,000/4,000/4,000	`aieng-lab/en-pronouns` (Drechsel & Herbold, 2026b)
PRONPERS	first, second, third	P	40,000/5,000/5,000	`aieng-lab/en-pronouns` (Drechsel & Herbold, 2026b)
IOI	indirect object (IO), subject	O	10,000/10,000/1,000	Source: `mib-bench/ioi` (Mueller et al., 2025); Processed `anonymous` (new)
KEYVALUE	value, other value	O	8,000/1,000/1,000	Controlled local generator; Processed `anonymous` (new)
INDUCTION	match, distractor	O	8,000/1,000/1,000	Controlled local generator; Processed `anonymous` (new)
REPETITION	yes, no	O	8,000/1,000/1,000	Controlled local generator; Processed `anonymous` (new)
FUNCCOMP	result, distractor	O	8,000/1,000/1,000	Controlled local generator; Processed `anonymous` (new)

Table 4: Representative labeled examples. Long natural contexts are shortened with an ellipsis. The masked span and local wording are retained. The right column gives the factual target token followed by its feature class. For one-sided tasks the counterfactual class is still listed in Table 3, but does not add a second base row.

Task	Masked context	Target / class
GENDER	Carl looked around the room, noticing tired expressions and disheartened faces. `[MASK]` realized then that something needed to change.	*he* / M
EMOTION	I am so unbelievably `[MASK]` for tomorrow.	*excited* / positive
RACE	Lutron alleged that products were made in Mexico and `[MASK]` and then imported into the United States.	*China* / Asian
RELIGION	The inscription should, from a `[MASK]` point of view, pinpoint where Jumne was located.	*Christian* / Christian
RAVELCONT	Toronto is a city in the continent of North America. Pilibhit is a city in the continent of `[MASK]`.	*Asia* / Asia
RAVELCNTRY	St. Petersburg is in Russia. Johnstown is in `[MASK]`.	*United States* / United States
RAVELLANG	People in Mexico City speak Spanish. People in Longreach speak `[MASK]`.	*English* / English
LANGID	`[en]` Every morning before work he quietly reads about a famous `[MASK]` in the paper.	*book* / English
PRONNUM	I was in it and around it. But certainly `[MASK]` had no corporeal existence.	*I* / singular
PRONPERS	Later editions may list both authors. `[MASK]` may also omit material or add new material.	*They* / third
IOI	As Carl and Maria left the consulate, Carl gave a fridge to `[MASK]`.	*Maria* / IO
KEYVALUE	`key5 = white; k20 = green; key5 = [MASK]`	*white* / value
INDUCTION	`j pear green j [MASK]`	*pear* / match
REPETITION	The operator's recent replies: yes yes yes yes `[MASK]`	*yes* / yes
FUNCCOMP	`m = fork; p = seal; h = m; h = [MASK]`	*fork* / result

Table 5: Task-dependent neutral dataset overview.

Task	Size	Source and Hugging Face ID
GENDER, RACE, RELIGION, IOI, KEYVALUE, INDUCTION, REPETITION, FUNCCOMP	4,593,588	`aieng-lab/biasneutral-ajibawa` (Drechsel & Herbold, 2026a)
EMOTION	1,000	`aieng-lab/en-sentiment-nrc-neutral` (Drechsel & Herbold, 2026b)
PRONNUM, PRONPERS	10,000	`aieng-lab/en-pronoun-neutral` (Drechsel & Herbold, 2026b)
RAVELCONT	5,000	Source: `hij/ravel`; Processed: `anonymous` (new)
RAVELCNTRY	5,000	Source: `hij/ravel`; Processed: `anonymous` (new)
RAVELLANG	5,000	Source: `hij/ravel`; Processed: `anonymous` (new)
LANGID	10,000	Source: `Helsinki-NLP/opus-100` and `facebookresearch/MUSE`; Processed: `anonymous` (new)

⇄IOI. This task is adapted from the Indirect Object Identification (IOI) task distributed with the Mechanistic Interpretability Benchmark (MIB; Mueller et al. (2025)). Each example contains an indirect object and a subject name. We use the indirect object as the factual target and the subject as the alternative target, yielding the one-sided IO-versus-subject feature used in our study. We retain the original MIB data split.

RAVEL. RAVELCONT, RAVELCNTRY, and RAVELLANG are adapted from the RAVEL benchmark (Huang et al., 2024), as distributed through MIB. They query the continent, country, or language associated with a city. For each attribute, we retain the three most frequent values in the RAVEL training split and sample an equal number of entities from each class. The city's true attribute value is the factual target, while a value from one of the other retained classes provides the alternative target. We remove city surface forms that map to multiple values of the queried attribute (e.g., *Amsterdam* in the Netherlands and the United States), collapse duplicate entities, and use each retained entity once with entity-disjoint splits. Consequently, the dataset sizes reflect distinct, unambiguous entities rather than the repeated query/intervention pairs in the original RAVEL data.

LANGID. This task is a three-way language-identification task with classes English, French, and German. We obtain monolingual sentences from OPUS bitext (Zhang et al., 2020; Tiedemann, 2012): English and French sentences are drawn from English–French data, and German sentences from English–German data. Only one side of each bitext pair is used, so every prediction context contains a single language. For each sentence, we mask a content word with at least ten words of left context. The original word is the factual target, while the alternative is its translation into one of the two rival languages obtained from the corresponding MUSE bilingual lexicon (Lample et al., 2018). We exclude stop words, short non-content candidates, words without a MUSE translation, identical cognates after normalization, and duplicate masked contexts. The resulting dataset is balanced across the three target languages.

KEYVALUE. This task is a symbolic retrieval task in which the model predicts the value associated with a queried key from two key–value bindings, e.g.,

`k1 = red; k2 = blue; k1 = [MASK].`

Keys are sampled without replacement from 30 manually specified identifiers (`k1–k20`, `key1–key5`, and `slot1–slot5`). Values are sampled without replacement from a manually curated vocabulary of 75 single-token colors, animals, and objects, with disjoint value vocabularies across train, validation, and test splits. The queried key is chosen from the two bindings: its associated value is the factual target, while the other value serves as the alternative target. All examples use the fixed canonical format shown above, with distinct keys and values and no duplicate masked contexts.

INDUCTION. This task is a sequence-completion task based on a repeated three-token unit. Three distinct tokens are sampled, the unit is repeated once, and its first token is appended as the query prefix; the model must then predict the second token of the unit. For example,

`j pear green j pear green j [MASK]`

has factual target *pear*. Tokens are drawn from a manually curated 50-token vocabulary consisting of the 26 lowercase Latin letters and 24 tokenizer-verified short words. The vocabulary is partitioned into disjoint train, validation, and test pools of 26, 12, and 12 tokens, respectively. The alternative is sampled either from another token in the unit or from three additional tokens outside the unit.

FUNCCOMP. This task is a symbolic alias-resolution task in which one variable aliases a variable holding the target value. Variable names are sampled without replacement from the 26 lowercase Latin letters, while values are drawn from the same split-disjoint 75-value vocabulary used for KEYVALUE. Examples contain one alias step and zero or one unrelated distractor assignment, using fixed canonical formatting. For example,

`m = fork; p = seal; h = m; h = [MASK]`

has factual target *fork*. If an unrelated assignment is present, its value serves as the alternative; otherwise, a different value is sampled from the split-local vocabulary. Alias and distractor variables are distinct, distractor values never equal the factual target, and duplicate masked contexts are removed.

Table 6: Language-model checkpoints and pretrained SAE resources used in the study.

Model	HF checkpoint	SAELens release	SAE ID / hook
Pythia-70M	`EleutherAI/pythia-70m-deduped`	`pythia-70m-deduped-res-sm`	`blocks.{L}.hook_resid_post`
GPT-2	`gpt2`	`gpt2-small-resid-post-v5-32k`	`blocks.{L}.hook_resid_post`
Llama-3.1-8B	`meta-llama/Llama-3.1-8B`	`llama_scope_lxr_32x`	`l{L}r_32x`

Neutral data. In addition to labeled task examples, the study uses neutral examples that do not express any of the feature classes of the corresponding task. They support both detection evaluation and intervention evaluation, including language modeling preservation measurements. Table 5 summarizes the neutral sources and their task assignments. For the semantic and grammatical tasks, we use existing neutral corpora matched to the corresponding feature domain. For the algorithmic tasks, general natural-language text serves as an out-of-task reference: it does not instantiate the symbolic computation being tested and therefore supports evaluation of feature specificity, while also providing ordinary language contexts for measuring preservation under intervention. For the RAVEL tasks, we instead construct domain-matched neutral examples from city prompts that query a different attribute, preserving the entity domain and prompt format without expressing the evaluated feature. For LANGID, neutral examples use Spanish, Italian, Dutch, and Portuguese sentences in the same input format as the labeled data; these languages are not among the task's three target classes.

B EXPERIMENTAL DETAILS

B.1 IMPLEMENTATION

We build on the `gradiend` Python package (Drechsel & Herbold, 2026b), extending its codebase with the generalized IEND implementations introduced in this work, and release this extended codebase.[1] The experiments are implemented in a separate study repository, which depends on this extended `gradiend` package and contains the task definitions, implementations of CAA, CGA, CAGA, and the SAE baselines, as well as the evaluation and analysis code.[2] Both repositories include documentation and configuration files for reproducing the experiments reported in this paper.

B.2 MODELS

We evaluate three pretrained decoder-only transformer language models (Vaswani et al., 2017) spanning different model families and scales. Table 6 lists the exact Hugging Face checkpoints and the pretrained SAELens resources used for the SAE reference.

B.3 COMPUTATIONAL COST AND MEMORY FOOTPRINT

Figure 5 reports the computational cost of one complete model-task-method evaluation cell for ⚥GENDER. For each method and model, this comprises task-specific feature fitting or selection, held-out detection evaluation, and intervention evaluation. For SAE, pretrained dictionary learning is excluded, while task-specific feature selection is included.

Importantly, these measurements characterize the evaluated implementations under the observed execution conditions rather than intrinsic algorithmic complexity, and should therefore be interpreted as rough runtime indicators. GPT-2 was run on an L40S and Llama-3.1-8B on an A100, so absolute runtime should only be compared within each model. The clearest runtime difference is between estimators: the IEND variants are consistently more expensive than their corresponding contrastive-mean methods, with particularly large increases for the gradient-based AGIEND and GRADIEND. For Llama-3.1-8B, SAE also incurs substantial runtime despite requiring no feature training, as its post-hoc procedure sweeps all layers, each with a substantially larger latent dictionary on a wider

[1] `anonymous`
[2] `anonymous`

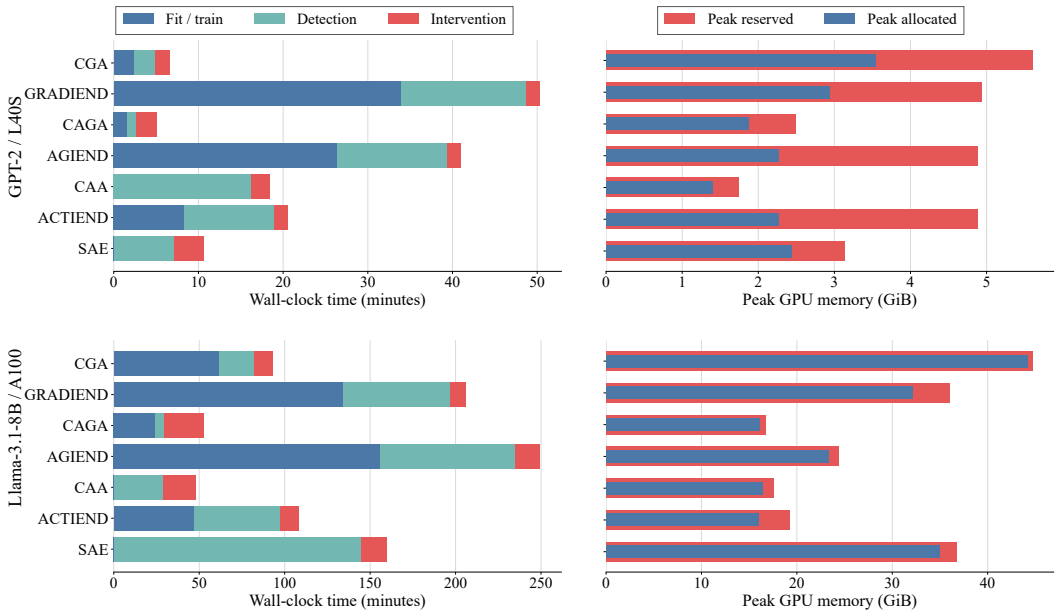


Figure 5: Computational cost and peak GPU memory on ⚥GENDER. Left: wall-clock time separated into fitting or training, detection, and intervention evaluation. Right: peak allocated and reserved CUDA memory across these stages. Reserved memory includes allocated memory and is therefore shown as an overlaid rather than additive quantity. SAE pretraining is excluded because we use pretrained dictionaries.

model. Peak memory follows the expected pattern, with parameter-gradient methods requiring substantially more memory than activation-space methods.

B.4 SIGNAL EXTRACTION

Let h_ℓ denote the residual stream output after transformer block ℓ and θ_ℓ the parameters of that block. For activation value and activation gradient signals, the single-layer and all-layer scopes are

$$\Omega^h_\ell = h_\ell, \qquad \Omega^h_{\text{all}} = (h_1, \ldots, h_L), \tag{2}$$

while for parameter-gradient signals, they are

$$\Omega^\theta_\ell = \theta_\ell, \qquad \Omega^\theta_{\text{all}} = (\theta_1, \ldots, \theta_L). \tag{3}$$

Embeddings and the final language-model head are excluded. For all-layer scopes, the corresponding layer-specific signals are concatenated in transformer-layer order.

Target spans. For each labeled example, let x denote the context preceding the feature-related target and let $y = (y_0, \ldots, y_{k-1})$ denote the target token span whose realization determines the corresponding feature class (e.g., a target "Asia" could be tokenized as y_0 ="Asi" and y_1 ="a"). Let $p, \ldots, p+k-1$, with $p \geq 1$, denote the sequence positions occupied by y. In a pairwise contrast, the same context x is paired with a factual target y^F and an alternative target y^A realizing different classes. In the definitions below, y denotes either candidate target.

Activation values. Activation value methods use the mean residual stream representation over the filled target span,

$$\phi^h_\ell(x, y) = \frac{1}{k} \sum_{j=0}^{k-1} h_{\ell, p+j}(x, y). \tag{4}$$

Thus, the activation value signal is extracted after the tokens realizing the feature have entered the causal context.

Table 7: Training, selection, and evaluation hyperparameters.

Setting	Value
IEND optimization	
GRADIEND learning rate	10^{-4} (GPT-2), 10^{-6} (Pythia-70M), 2×10^{-6} (Llama-3.1-8B)
ACTIEND encoder learning rate	5×10^{-5} (Pythia-70M, GPT-2), 10^{-5} (Llama-3.1-8B)
ACTIEND decoder learning rate	1.15×10^{-2}
AGIEND learning rate	3×10^{-3} (Pythia-70M, GPT-2, Llama-3.1-8B)
Training steps	500 (GRADIEND, AGIEND); 100 (ACTIEND)
Evaluation interval	50 steps (GRADIEND, AGIEND); 20 steps (ACTIEND)
Training batch size	8
Maximum sequence length	128 tokens
Random seeds	Up to 3, starting from seed 0; the first converged run is retained
Parameter-gradient methods	
GRADIEND pre-pruning	16 examples; retain top 10% of coordinates (GPT-2, Pythia-70M) or top 1% (Llama-3.1-8B)
GRADIEND post-pruning	Retain top 1% of decoder weights
CGA coordinate scope	Full parameter gradient (GPT-2, Pythia-70M); same pre-pruned coordinates as GRADIEND (Llama-3.1-8B)
Sampling and selection	
Neutral-example caps	1,000 training / 300 validation / 1,000 test
Labeled encoder-evaluation caps	50 training / 200 validation examples per class
Intervention-evaluation caps	200 validation / 1,000 test examples per class
SAE latent eligibility	Activation on at least 1% of labeled validation examples
Intervention selection	
Candidate magnitudes	10^{-5}, 2×10^{-5}, 5×10^{-5}, 10^{-4}, $\ldots$, 10^{5} (31 values)
Preservation constraint	$\mathrm{LMS}(I) \geq 0.99\,\mathrm{LMS}(I_0)$

Gradient signals. Following the original GRADIEND formulation (Drechsel & Herbold, 2026a), gradient-based methods condition on the first token y_0 of the target through the causal-language-modeling loss

$$\mathcal{L}_{\mathrm{CLM}}(x, y) = -\log p_\theta(y_0 \mid x). \tag{5}$$

The activation gradient signal at layer ℓ is the gradient of this loss with respect to the residual stream at the preceding prediction position,

$$\phi_\ell^{\nabla h}(x, y) = \nabla_{h_{\ell,p-1}} \mathcal{L}_{\mathrm{CLM}}(x, y). \tag{6}$$

Parameter gradient methods use the same objective and compute

$$\phi_\ell^{\nabla \theta}(x, y) = \nabla_{\theta_\ell} \mathcal{L}_{\mathrm{CLM}}(x, y). \tag{7}$$

Activation-steering positions. Following common activation-steering practice (Turner et al., 2025; Rimsky et al., 2024), all activation-space interventions are applied across all prefix token positions, rather than only at the position immediately preceding the target.

Target-position asymmetry. Because factual and alternative targets in a contrast share the same preceding context, their residual stream representations at the prediction position $p-1$ are identical. activation value contrasts are therefore extracted from the filled target span, whereas gradient signals condition on the first target token through the next-token loss at $p-1$.

B.5 Training and selection

Table 7 summarizes the training, sampling, and selection hyperparameters. The following sections provide the corresponding method-specific details.

IEND convergence. For pairwise IEND, convergence is assessed using the correlation between the encoder output and the desired labels, assigning the two contrasted classes -1 and $+1$ and neutral examples 0. This follows the established GRADIEND workflow, which has so far considered pairwise tasks only (Drechsel & Herbold, 2026b). For the newly introduced one-sided setting, we instead use AUROC for distinguishing the target class from the pooled non-target examples. This avoids rewarding separation between neutral examples and alternative target classes, which is irrelevant to the one-sided objective.

Representation selection. For SAE, CAA, CGA, and CAGA, the candidate set comprises all available individual layer scopes and the corresponding all-layer scope defined in Section B.4. For each evaluated feature, we select the scope $\Omega^\star$ with the highest validation $\mathrm{Det}.$. The selected scope is then fixed for both held-out detection and intervention.

We do not perform the analogous layer sweep for the IEND methods. For the contrastive mean methods, once the layerwise signals have been extracted, additional scopes can be evaluated by applying the same mean-difference estimator to the corresponding signals. For IEND, each candidate scope instead requires training a separate encoder-decoder representation. We therefore use the all-layer scope throughout: $\Omega^{\theta}_{\mathrm{all}}$ for GRADIEND and $\Omega^{h}_{\mathrm{all}}$ for ACTIEND and AGIEND.

Evaluation samples. Sampling limits are applied per feature class and are summarized in Table 7. Neutral examples are sampled independently from the corresponding training, validation, and test pools. Neutral training examples are used only by the IEND methods (GRADIEND, ACTIEND, and AGIEND), where neutral texts form identity transitions with zero desired change and thereby regularize the learned representation toward neutral behavior.

Validation neutrals are used for detection-based selection and threshold estimation and, for SAE, additionally enter latent ranking through the neutral-specificity term. They also provide the preservation set used when selecting intervention strengths. Test neutrals are reserved for final evaluation

Threshold selection. Classification thresholds are selected on validation data by maximizing Youden's $J = \mathrm{TPR} + \mathrm{TNR} - 1$. For Spec_n, we fit a target-versus-neutral decision rule. For Excl, we fit a separate target-versus-rival decision rule for each rival class. For each rule, both score orientation and threshold are determined on validation data and then applied unchanged to the held-out test split.

SAE selection. Following prior supervised approaches that identify concept-relevant SAE latents from class-conditional activations (Karvonen et al., 2025; Gallifant et al., 2025), we rank eligible latents on the validation split. For target class c, layer ℓ, and SAE latent j, the ranking score is

$$s_{\ell cj} = \bar{z}_{\ell cj} - \bar{z}_{\ell,\neg c,j} - \left|\bar{z}_{\ell,N,j}\right|, \tag{8}$$

where $\bar{z}_{\ell cj}$ is the mean latent activation on examples of class c, $\bar{z}_{\ell,\neg c,j}$ the mean over labeled non-c examples, and $\bar{z}_{\ell,N,j}$ the mean over neutral examples. Latents firing on fewer than 1% of labeled validation examples are ineligible.

Eligible latents are ranked separately for each feature class and layer. An individual-layer candidate uses the top-ranked eligible latent at that layer. For the all-layer candidate, we independently select the top-ranked eligible latent at every available layer and sum their activations without weighting. These candidates then enter the representation-selection procedure described above.

Parameter-gradient pruning. GRADIEND uses the pre- and post-pruning procedure introduced with the `gradiend` package (Drechsel & Herbold, 2026b), with the settings reported in Table 7. CGA uses the full parameter-gradient signal for GPT-2 and Pythia-70M, and the same pre-pruned coordinate set as GRADIEND for Llama-3.1-8B.

B.6 Intervention evaluation

The candidate intervention magnitudes and preservation constraint used for validation-based intervention selection are summarized in Table 7.

For SAE, intervention follows the decoder directions corresponding to the latents used by the selected representation. For a single layer representation, we use the decoder direction of the selected latent at that layer. For an all-layer representation, the selected decoder directions are applied simultaneously at their corresponding residual stream layers.

B.7 Aggregation

For each method-model-task combination, we compute Det. and Int. for every evaluated target-class and average these scores with equal weight. For pairwise constructions with K classes, both

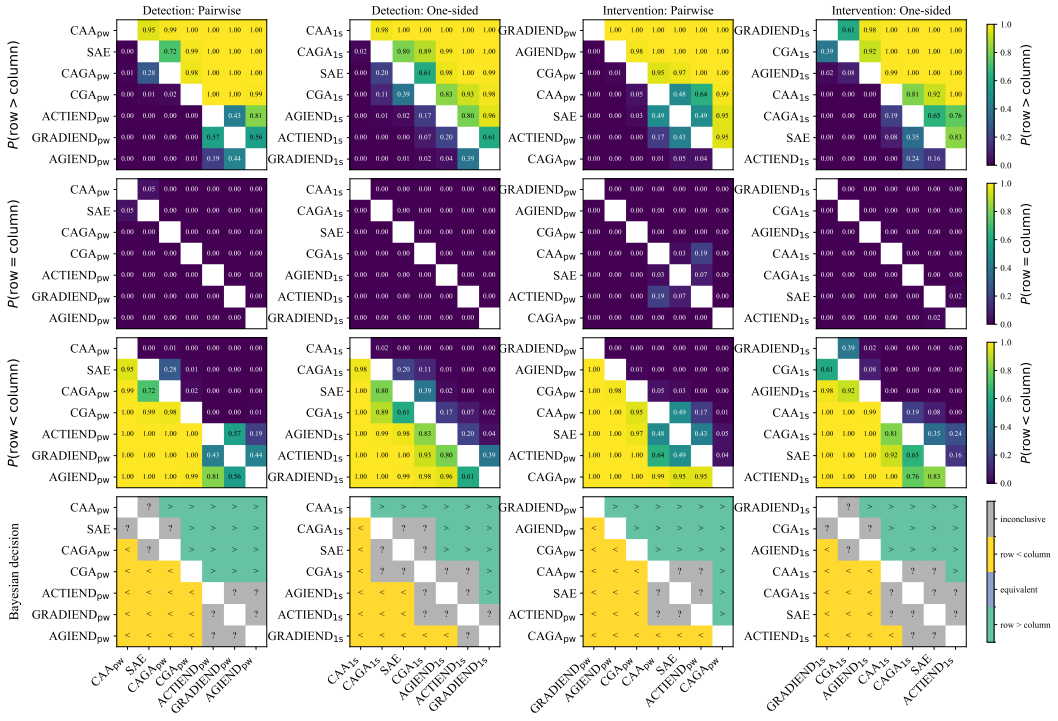


Figure 6: Bayesian pairwise comparison of methods.

endpoints of each unordered class pair are evaluated, yielding $2\binom{K}{2}$ scores. One-sided constructions and SAE yield one score per target class.

For higher-level summaries, task scores are averaged equally within each model, and the resulting model-level means are averaged equally across models.

B.8 Statistical analysis

We compare methods using the Bayesian analysis implemented in AutoRank (Herbold, 2020). For detection and intervention separately, each matched model–task combination is considered one observation, using the corresponding aggregated task-level score defined in Section B.7. Thus, individual examples, feature classes, and pairwise contrasts are not treated as independent observations.

We use an absolute region of practical equivalence (ROPE) of 0.01 in the respective metric scale. A pairwise outcome is assigned when the posterior probability of one of the three decisions (the first method performing better, the second performing better, or practical equivalence) exceeds $1 - \alpha = 0.95$; otherwise, the comparison remains unresolved. Posterior probabilities are estimated from 50,000 samples.

C Detailed results

This section provides additional analyses supporting and extending the results in Section 5.

C.1 Statistical analysis

Figure 6 reports the full Bayesian pairwise comparison underlying the summary shown in Figure 4b, including the posterior probabilities that the row method performs better, performs worse, or is practically equivalent to the column method, together with the resulting Bayesian decision.

Table 8: Effect of representation scope. Single-layer representations are selected by validation Detection. Δ denotes single layer minus all-layer performance on the held-out test split. Win rates report the fraction of class/contrast comparisons with $\Delta > 0$, i.e., where the selected single layer outperforms the all-layer representation on the held-out test split.

	Selected	Detection		Intervention	
Method	Single layer	Win	Mean Δ	Win	Mean Δ
SAE	96.9%	49.0%	+0.035	27.3%	−0.013
CAA	87.9%	82.5%	+0.082	45.0%	−0.001
CGA	95.4%	85.0%	+0.153	34.8%	−0.041
CAGA	94.2%	86.3%	+0.140	38.3%	−0.015
Overall	93.0%	80.4%	+0.115	38.1%	−0.018

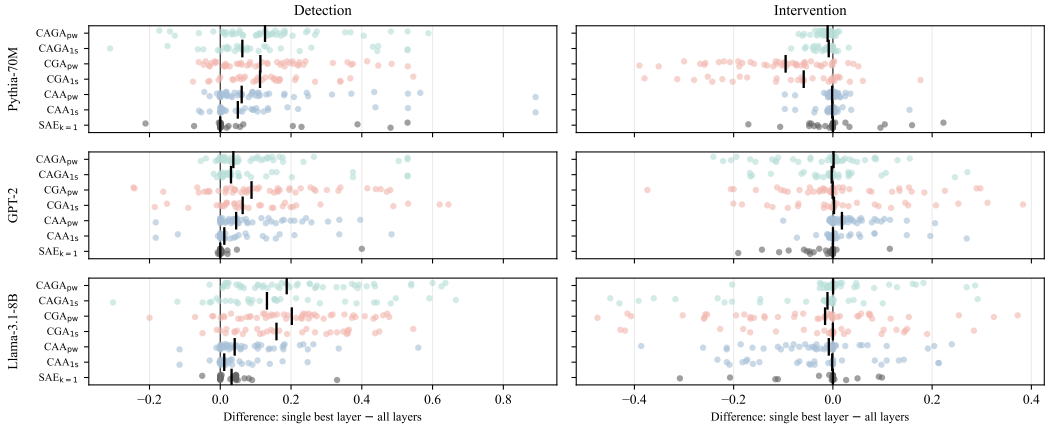


Figure 7: Difference between the validation-selected best single-layer and all-layer representations for detection and intervention. Positive values indicate higher performance for the selected single layer.

C.2 Layer localization and representation scope

For SAE, CAA, CGA, and CAGA, the main experiment selection procedure considers both the predefined all-layer representation and every available single best layer representation, and selects among them by validation Det.. To isolate the effect of representation scope, we additionally compare the all-layer candidate with the single layer achieving the highest validation Det., separately for each target class or pairwise contrast.

Single-layer versus all-layer representations. Table 8 summarizes the comparison. Validation detection strongly favors single layers over the all-layer representation. For the targeted contrastive mean methods, this preference also generalizes to held-out detection, with particularly large gains for CGA and CAGA. For SAE, in contrast, the strong validation preference for a single layer does not translate into a consistent held-out advantage.

Figure 7 shows the corresponding distribution of single-layer minus all-layer differences for detection and intervention. Intervention does not exhibit the same localization advantage. Thus, the layer best suited to detecting a feature is not generally the scope best suited to intervening on it.

Robustness of the main comparison. To assess whether representation-scope selection affects the main comparison, Figure 8 repeats Figure 3 with SAE and contrastive mean methods restricted to their all-layer representations. The qualitative conclusions remain unchanged.

Layerwise profiles. Figures 9–10 report validation detection and intervention across single layers and the all-layer representation for each task on the two smaller models, Pythia-70M and

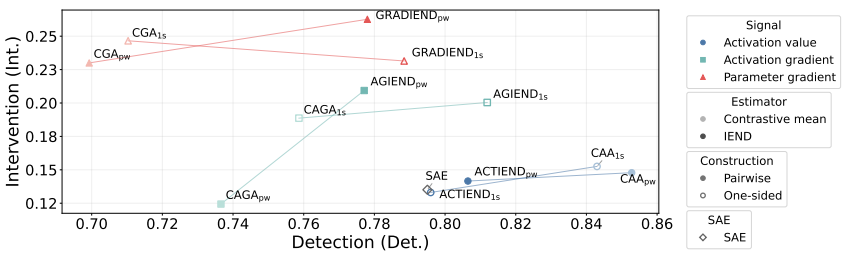


Figure 8: All-layer robustness analysis. Detection-intervention comparison corresponding to Figure 3, with SAE and contrastive mean methods restricted to their all-layer representations rather than using validation-selected scopes.

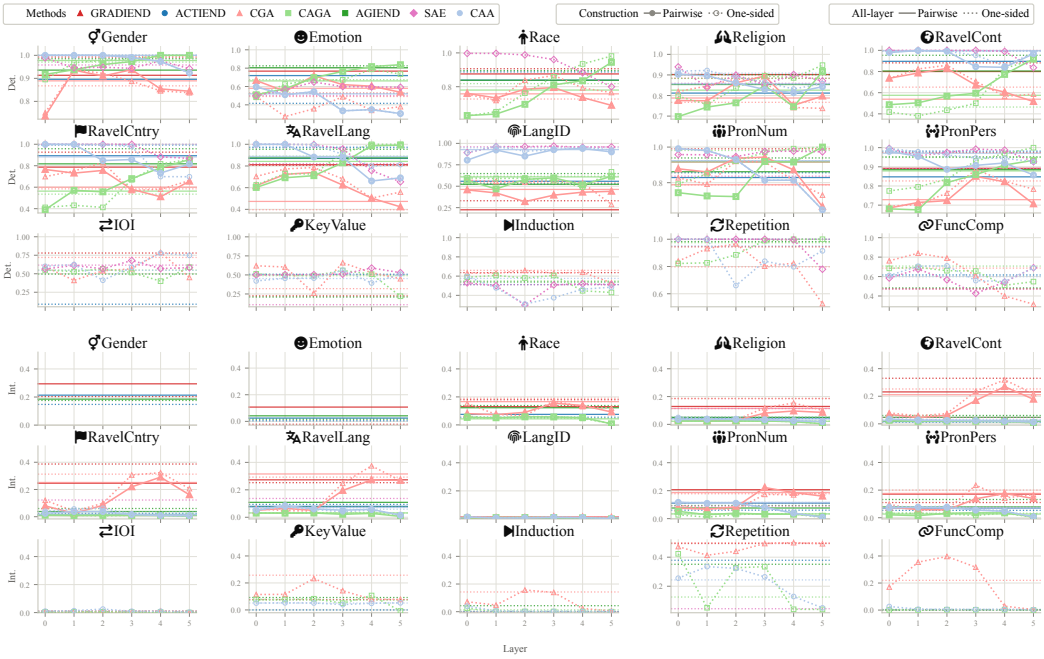


Figure 9: Layerwise detection and intervention for Pythia-70M.

GPT-2. We restrict this exhaustive layerwise analysis to these models for computational tractability: Llama-3.1-8B exposes substantially more layers, making a full per-task layer sweep considerably more expensive. The profiles vary substantially across tasks and methods, and layers with strong detection do not necessarily yield strong intervention.

C.3 Sparse autoencoder variants

Activation value methods face a positional asymmetry between detection and intervention. Remember that distinguishing the factual and alternative targets requires observing the filled target representation, whereas an intervention intended to affect that target must act before it is generated. Because the pretrained SAE is learned independently of the requested feature, we can additionally select the relevant latent from the predictive representation immediately preceding the target. We denote this variant by $\mathrm{SAE}_{\mathrm{pre}}$.

A feature may also be distributed across multiple SAE latents. We therefore compare the single latent representation ($k = 1$) used in the main experiments with a validation selected latent set $k^\star$. At the selected layer, we sum the activations of the top-k ranked eligible latents for

$$k \in \{1, 2, 4, 8, 16, 32, 64, 128\}.$$

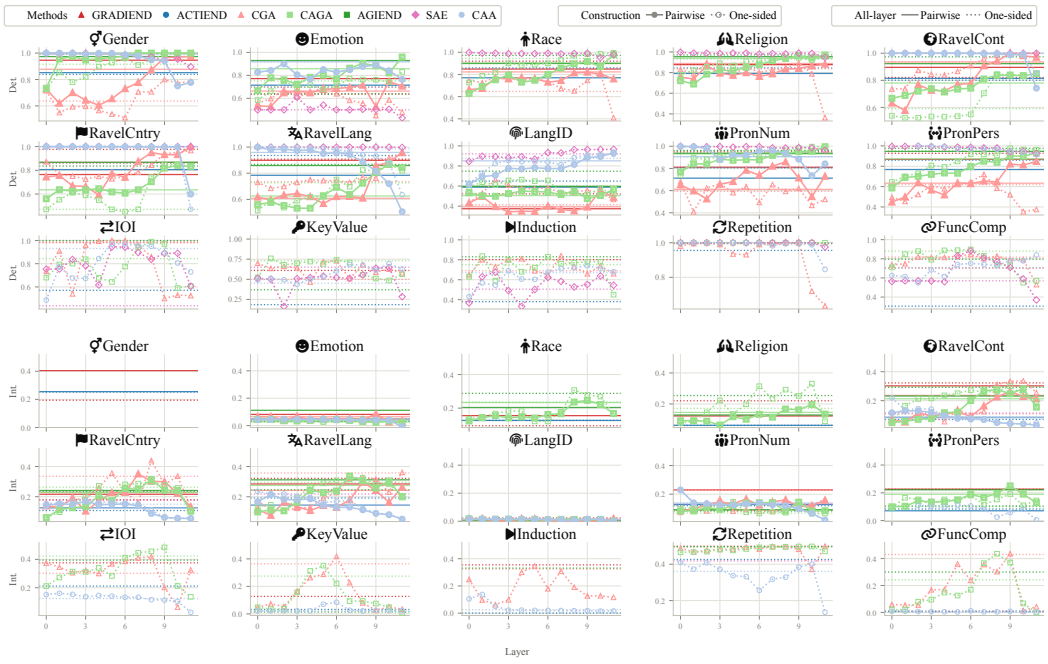


Figure 10: Layerwise detection and intervention for GPT-2.

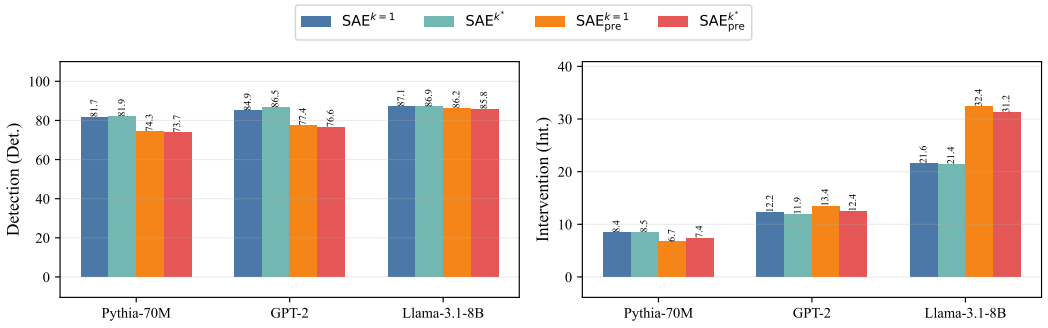


Figure 11: Detection and Intervention for the four SAE variants obtained by varying latent-selection position (filled-target vs. pre-target) and the number of selected latents ($k = 1$ vs. $k^\star$).

We select $k^\star$ on validation data by maximizing $\mathrm{AUC}_n + \mathrm{AUC}_o$ when a rival class is defined and AUC_n otherwise, resolving ties in favor of smaller k. The same procedure is applied to $\mathrm{SAE}_{\mathrm{pre}}$.

Together, the positional and latent-count choices yield four variants: $\mathrm{SAE}_{k=1}$, $\mathrm{SAE}_{k^\star}$, $\mathrm{SAE}_{\mathrm{pre},k=1}$, and $\mathrm{SAE}_{\mathrm{pre},k^\star}$. Figure 11 compares their detection and intervention performance.

Increasing the number of selected latents from $k = 1$ to $k^\star$ changes both detection and intervention only modestly across models. The representation position has a substantially larger effect: $\mathrm{SAE}_{\mathrm{pre}}$ yields lower detection than the corresponding filled-target variant on all thre models. For intervention, the effect is model-dependent; pre-target selection provides no advantage on Pythia-70Mor GPT-2,but produces a pronounced increase on Llama-3.1-8B. Thus, selecting features before the target does not provide a consistent overall advantage. We therefore retain the filled-target $\mathrm{SAE}^{k=1}$ as the main reference.

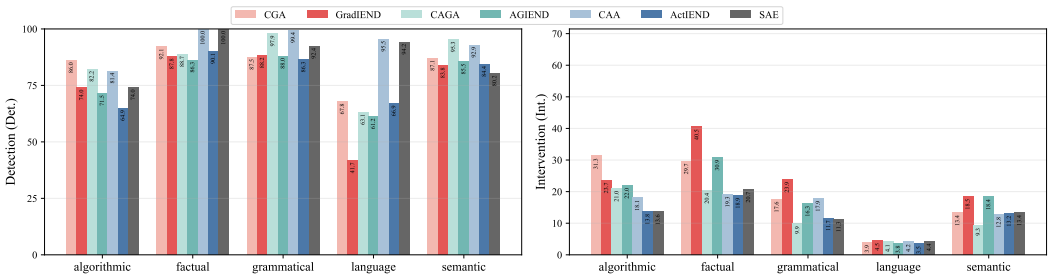


Figure 12: Performance by task family averaged across models.

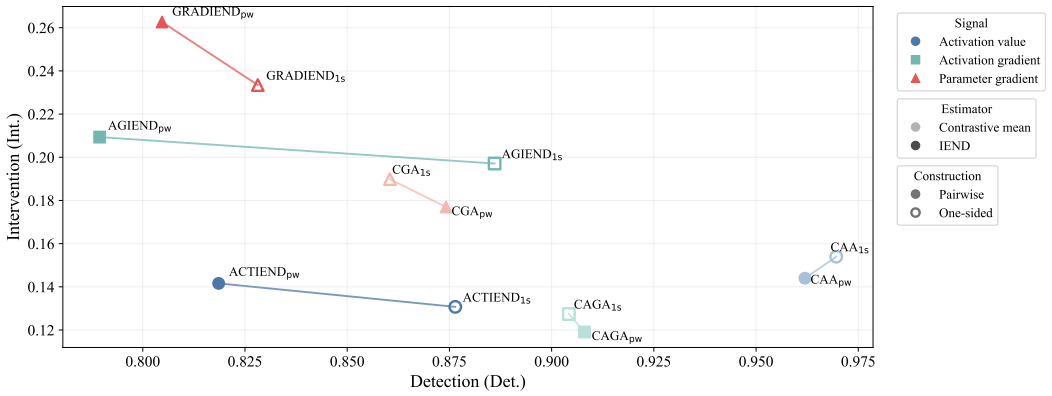


Figure 13: **Pairwise versus one-sided feature construction.** Model-balanced mean Detection and Intervention on tasks supporting both constructions.

C.4 Task characteristics

Figure 12 aggregates performance by task family (as introduced in Table 2). Detection is generally high for factual, grammatical, and semantic tasks, whereas the language task reveals substantially greater differences between methods: CAA and SAE retain high detection, while several targeted methods drop considerably. Intervention varies substantially across task families. Factual tasks produce the largest effects for most methods, whereas language tasks consistently yield only small interventions. Semantic and algorithmic tasks generally fall between these extremes, while grammatical tasks tend to show somewhat weaker effects.

C.5 Pairwise and one-sided feature specifications

To compare pairwise and one-sided construction directly, we restrict the analysis to tasks for which both specifications are available (excluding ⇄IOI, 🔁Repetition, 🔗FuncComp, 🔑KeyValue, ⏭Induction). Figure 13 shows that one-sided construction often achieves higher detection than pairwise construction, although the differences are modest for most methods. Intervention is often similar except for GRADIEND. Overall, one-sided construction usually at least matches the performance of pairwise construction while requiring only K rather than $\binom{K}{2}$ feature models and remaining applicable when natural examples are available only for the target class.

C.6 IEND optimization across signal spaces

Unlike the contrastive mean estimators, IEND requires optimization choices such as learning rate and number of training steps. For GRADIEND and ACTIEND, we ran a short learning-rate screen on ⚥Gender for each model and retained the largest rate that converged under the respective fixed step budget. The resulting rates were then fixed across all tasks.

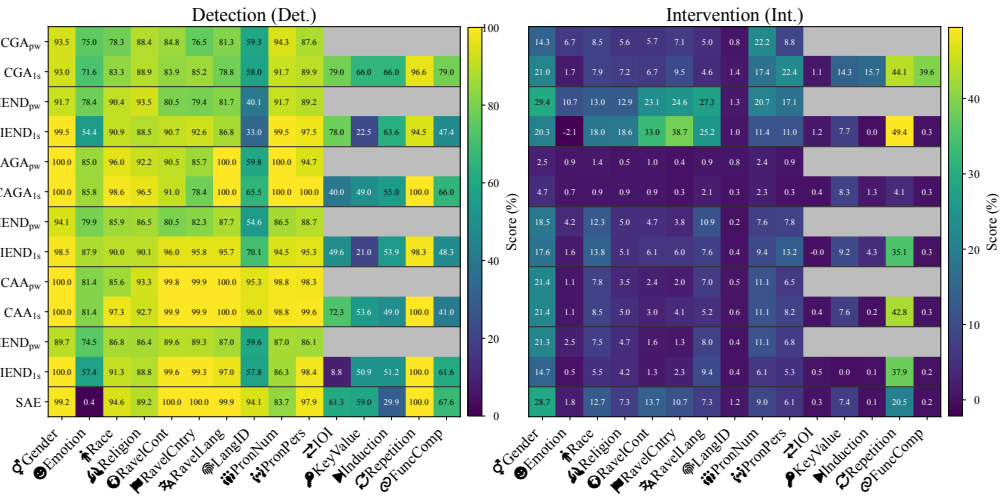


Figure 14: Pythia-70M detection and intervention scores per task.

Decoder learning rate. The original GRADIEND formulation (Drechsel & Herbold, 2026a) jointly optimizes encoder and decoder with a shared learning rate, which we initially also used for the other IEND variants. For ACTIEND, however, increasing only the decoder learning rate while keeping the encoder rate fixed substantially improved intervention in experiments. Analogous decoder-rate increases did not yield a clear benefit for GRADIEND or AGIEND. We therefore use a larger decoder than encoder learning rate for ACTIEND, while GRADIEND and AGIEND retain shared rates.

Optimization steps. We also experimented with shorter optimization schedules together with corresponding learning-rate adjustments. For ACTIEND, reducing the budget from 500 to 100 steps while increasing the decoder learning rate by a factor of five produced similar detection and intervention in development experiments. Analogous shorter schedules for GRADIEND and AGIEND led to weaker intervention. We therefore use 100 steps for ACTIEND and 500 for the gradient-based variants.

Scope. The experiments described in this section are small, single-model, single-task ablations intended to motivate the optimization choices rather than to establish general laws. They support treating the encoder and decoder learning rates as signal-dependent IEND parameters, instead of assuming that the shared schedule of GRADIEND transfers unchanged to other signal spaces.

C.7 Complete task-level results

For completeness, we report the task-level quantities underlying all aggregate results in Tables 9–14. Further, we provide model-based views in Figures 14-16 of the averaged version in Figure 4a.

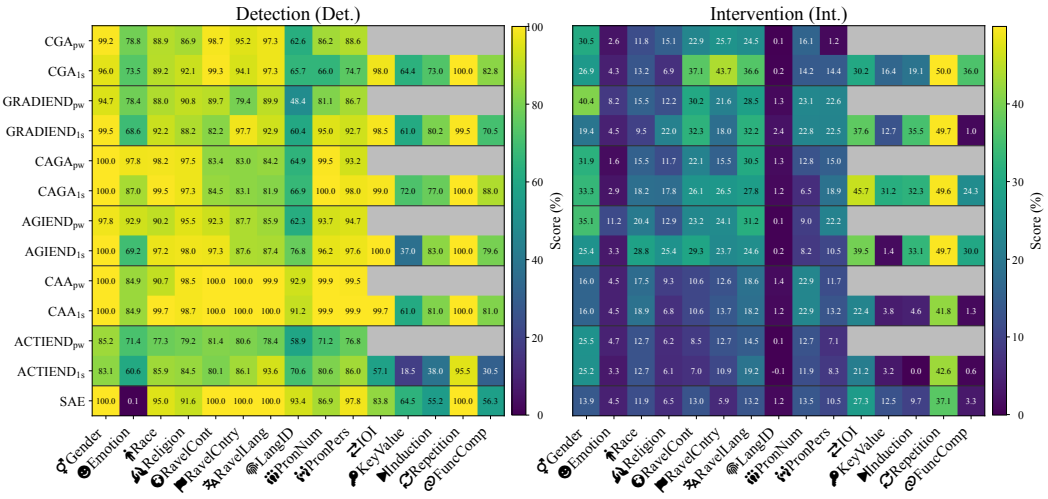


Figure 15: GPT-2 detection and intervention scores per task.

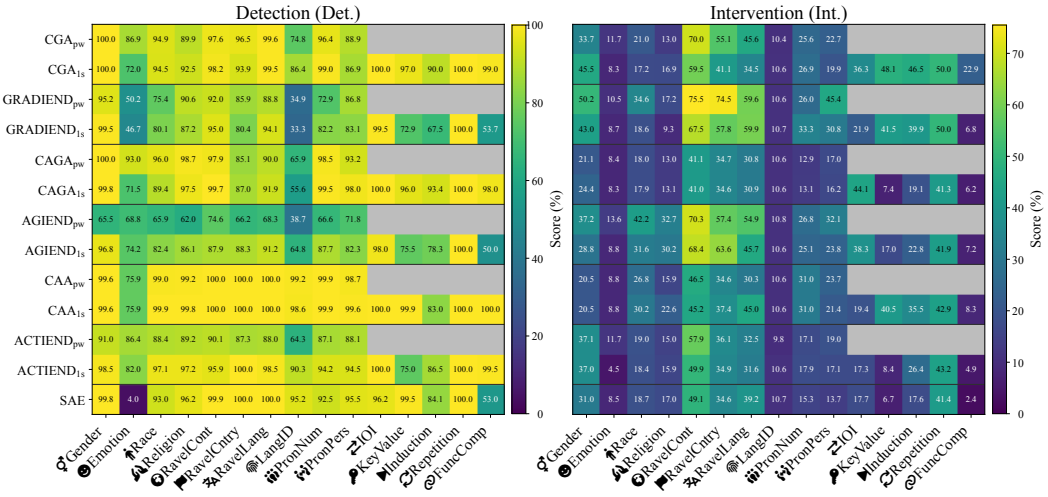


Figure 16: Llama-3.1-8B detection and intervention scores per task.

Table 9: Detection raw results for Pythia-70M.

Method	Gender	Emotion	Race	Religion	RavelCont	RavelCntry	RavelLang	LangID	PronNum	PronPers	IOI	KeyValue	Induction	Repetition	FuncComp
							AUC_n								
$\mathrm{CGA}_{\mathrm{pw}}$	0.966	0.783	0.866	0.911	0.931	0.914	0.836	0.593	0.980	0.900	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.951	0.687	0.891	0.907	0.887	0.856	0.808	0.604	0.977	0.899	0.867	0.725	0.701	0.966	0.990
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.997	**0.940**	0.952	0.960	0.953	0.955	0.947	0.508	0.994	0.997	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	0.997	0.545	0.919	0.910	0.925	0.968	0.868	0.541	0.995	0.989	0.827	0.543	0.649	0.999	0.474
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.851	0.966	0.966	0.974	0.857	**1.000**	0.767	**1.000**	0.977	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.893	**0.993**	0.972	0.963	0.927	**1.000**	0.735	**1.000**	**1.000**	0.956	0.628	0.631	**1.000**	0.891
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.996	0.938	0.958	0.963	0.987	0.987	0.995	0.729	0.982	0.990	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	0.989	0.879	0.933	0.932	0.974	0.992	0.957	0.740	0.976	0.953	0.605	0.423	0.816	0.983	0.927
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.813	0.897	**0.979**	0.999	**1.000**	1.000	0.980	1.000	0.997	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.813	0.992	0.959	1.000	1.000	1.000	0.980	1.000	0.997	0.733	0.778	0.490	**1.000**	0.913
$\mathrm{ACTIEND}_{\mathrm{pw}}$	1.000	0.914	0.960	0.924	0.999	0.998	0.994	0.801	0.939	0.967	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	**1.000**	0.731	0.985	0.888	0.999	0.994	0.992	0.772	0.863	0.994	0.506	0.660	**0.999**	**1.000**	0.702
SAE	1.000	0.502	0.959	0.954	**1.000**	1.000	1.000	**0.983**	0.841	0.996	**1.000**	**0.999**	0.988	**1.000**	**1.000**
							AUC_o								
$\mathrm{CGA}_{\mathrm{pw}}$	0.995	0.920	0.857	0.902	0.885	0.859	0.813	0.594	**1.000**	0.898	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.992	0.776	0.920	0.956	0.891	0.852	0.828	0.580	1.000	0.938	0.903	**0.660**	0.710	0.998	**0.841**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.977	0.986	0.992	0.962	0.992	0.962	0.517	**1.000**	**1.000**	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	**1.000**	0.558	0.968	0.983	0.944	0.980	0.888	0.503	1.000	0.999	0.856	0.491	**0.741**	0.989	0.564
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**0.981**	0.960	0.969	0.912	0.998	**1.000**	0.677	**1.000**	0.984	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.968	0.986	0.965	0.985	0.874	**1.000**	0.655	**1.000**	1.000	0.645	0.636	0.643	**1.000**	0.707
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.995	0.972	0.990	0.989	1.000	**1.000**	1.000	0.787	0.995	0.996	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	0.996	0.936	0.933	0.975	0.993	**1.000**	0.982	0.761	0.985	0.978	0.496	0.487	0.539	0.995	0.483
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.933	0.856	0.933	**1.000**	**1.000**	**1.000**	0.983	**1.000**	0.997	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.933	**0.999**	0.931	**1.000**	**1.000**	**1.000**	**0.986**	**1.000**	**1.000**	**0.976**	0.536	0.490	**1.000**	0.692
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	0.975	0.996	**0.993**	**1.000**	**1.000**	**1.000**	0.818	1.000	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	**1.000**	0.770	0.969	0.973	**1.000**	**1.000**	**1.000**	0.743	0.998	1.000	0.586	0.509	0.522	**1.000**	0.616
SAE	0.992	0.500	0.946	0.892	**1.000**	**1.000**	**1.000**	0.966	0.837	0.992	0.613	0.590	0.527	**1.000**	0.692
							Spec_n								
$\mathrm{CGA}_{\mathrm{pw}}$	0.950	0.718	0.868	0.920	0.939	0.925	0.882	0.750	0.943	0.876	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.953	0.575	0.892	0.938	0.923	0.913	0.908	0.825	0.917	0.913	0.915	0.950	0.945	0.990	**1.000**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.913	0.767	0.904	0.935	0.805	0.794	0.817	0.401	0.917	0.892	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	0.992	0.598	0.955	0.965	0.915	0.926	0.922	0.707	**1.000**	0.975	0.920	0.850	0.844	0.990	0.900
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.853	0.981	**0.978**	0.947	0.948	**1.000**	0.866	**1.000**	0.983	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.913	**1.000**	0.967	0.910	0.882	**1.000**	0.948	**1.000**	**1.000**	0.900	0.935	0.794	**1.000**	0.955
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.943	0.805	0.859	0.865	0.805	0.823	0.877	0.546	0.865	0.887	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	0.988	0.832	0.948	0.942	0.960	0.958	0.965	0.708	0.945	0.953	0.965	0.920	0.910	0.990	0.960
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.876	0.988	0.943	0.998	0.999	1.000	0.953	0.988	0.983	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.876	0.973	0.927	0.999	0.999	1.000	0.960	0.988	0.996	0.723	0.947	0.779	**1.000**	0.867
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.896	0.739	0.868	0.864	0.896	0.893	0.870	0.596	0.870	0.861	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	1.000	0.878	0.966	0.888	0.996	0.993	0.970	0.578	0.960	0.984	0.088	0.539	**0.991**	**1.000**	0.747
SAE	1.000	**0.998**	1.000	0.975	**1.000**	**1.000**	0.999	**0.993**	0.998	0.999	**0.991**	**0.991**	0.957	**1.000**	0.999
							Excl.								
$\mathrm{CGA}_{\mathrm{pw}}$	0.945	0.815	0.783	0.884	0.848	0.765	0.818	0.732	**1.000**	0.924	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.980	0.715	0.833	0.889	0.839	0.922	0.788	0.798	0.995	0.934	0.790	0.670	0.660	0.970	0.790
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.966	0.957	0.935	0.965	0.988	0.956	0.424	**1.000**	**1.000**	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	**1.000**	0.591	0.909	0.885	0.907	0.963	0.924	0.330	0.995	0.986	0.780	0.225	0.636	0.945	**0.910**
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.925	0.975	0.922	0.905	0.980	**1.000**	0.598	**1.000**	0.947	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.915	0.990	0.970	0.976	0.784	**1.000**	0.783	**1.000**	**1.000**	0.400	0.490	0.550	**1.000**	0.660
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.989	0.925	0.980	0.954	0.999	**1.000**	0.998	0.757	0.984	0.994	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	0.990	0.888	0.900	0.901	**1.000**	**1.000**	**1.000**	0.701	0.952	0.970	0.545	0.210	0.585	0.985	0.765
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.770	0.924	**0.990**	**1.000**	**1.000**	**1.000**	0.982	**1.000**	0.990	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.770	**1.000**	0.965	**1.000**	**1.000**	**1.000**	**0.990**	**1.000**	**1.000**	**0.950**	0.700	**0.940**	**1.000**	0.410
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	0.890	0.977	0.957	**1.000**	**1.000**	**1.000**	0.750	0.995	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	**1.000**	0.416	0.913	0.911	**1.000**	**1.000**	**1.000**	0.727	0.965	0.997	0.390	**0.790**	0.512	**1.000**	0.650
SAE	1.000	**1.000**	**1.000**	0.971	**1.000**	**1.000**	**1.000**	0.941	0.968	0.979	0.668	0.625	0.299	**1.000**	0.676
							Det.								
$\mathrm{CGA}_{\mathrm{pw}}$	0.942	0.698	0.766	0.866	0.838	0.747	0.737	0.566	0.943	0.853	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.930	0.503	0.833	0.852	0.832	0.823	0.773	0.563	0.917	0.897	**0.790**	**0.660**	**0.660**	0.966	**0.790**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.913	0.767	0.876	0.902	0.800	0.787	0.812	0.225	0.917	0.892	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	0.992	0.527	0.906	0.884	0.882	0.926	0.807	0.330	0.990	0.974	0.780	0.225	0.636	0.945	0.474
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.817	0.937	0.917	0.864	0.857	**1.000**	0.573	**1.000**	0.943	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	**0.857**	**0.981**	**0.951**	0.905	0.784	**1.000**	0.621	**1.000**	**1.000**	0.400	0.490	0.550	**1.000**	0.660
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.936	0.805	0.838	0.854	0.805	0.818	0.875	0.523	0.862	0.885	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	0.985	0.825	0.896	0.901	0.954	0.958	0.954	0.648	0.939	0.952	0.496	0.210	0.539	0.983	0.483
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.716	0.808	0.904	0.998	0.999	1.000	0.947	0.988	0.976	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.716	0.973	0.911	0.999	0.999	1.000	**0.957**	0.988	0.996	0.723	0.536	0.490	**1.000**	0.410
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.896	0.719	0.839	0.812	0.896	0.893	0.870	0.558	0.829	0.848	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	1.000	0.416	0.887	0.855	0.996	0.993	0.970	0.529	0.856	0.981	0.088	0.509	0.512	**1.000**	0.616
SAE	0.992	0.500	0.945	0.892	**1.000**	**1.000**	0.999	0.936	0.837	0.974	0.613	0.590	0.299	**1.000**	0.676

Table 10: Intervention raw results for Pythia-70M.

Method	GENDER	EMOTION	RACE	RELIGION	RAVELCONT	RAVELCNTRY	RAVELLANG	LANGID	PRONNUM	PRONPERS	IOI	KEYVALUE	INDUCTION	REPETITION	FUNCCOMP
							ΔP^+								
CGA_{pw}	0.197	0.084	0.074	0.065	0.078	0.121	0.044	0.000	**0.331**	0.104	-	-	-	-	-
CGA_{1s}	0.332	0.004	0.036	0.091	0.097	0.169	0.030	0.000	0.244	**0.321**	0.012	**0.126**	**0.309**	0.106	**0.787**
$\text{GRADIEND}_{\text{pw}}$	**0.499**	**0.171**	0.144	0.205	0.425	0.471	**0.482**	**0.002**	0.327	0.244	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.296	0.001	**0.237**	**0.318**	**0.621**	**0.753**	0.438	0.000	0.119	0.092	**0.014**	0.000	0.000	**0.140**	0.001
CAGA_{pw}	0.004	0.002	0.002	0.000	0.003	0.002	0.002	0.000	0.005	0.004	-	-	-	-	-
CAGA_{1s}	0.001	-0.000	0.002	0.002	0.003	0.000	0.015	0.000	0.007	0.002	0.001	0.016	0.026	0.005	0.000
$\text{AGIEND}_{\text{pw}}$	0.237	0.041	0.152	0.069	0.079	0.063	0.158	0.000	0.100	0.083	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.216	0.004	0.166	0.069	0.117	0.102	0.092	-0.000	0.144	0.159	-0.000	0.028	0.085	0.135	-0.000
CAA_{pw}	0.248	-0.009	0.037	0.019	0.013	0.019	0.076	0.000	0.118	0.043	-	-	-	-	-
CAA_{1s}	0.248	-0.009	0.050	0.049	0.024	0.075	0.044	-0.000	0.118	0.064	-0.000	-0.002	-0.000	0.029	0.001
$\text{ACTIEND}_{\text{pw}}$	0.290	0.020	0.048	0.052	0.015	0.017	0.093	0.001	0.123	0.049	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.187	0.000	0.070	0.048	0.013	0.036	0.125	-0.000	0.079	0.051	0.002	-0.000	0.000	0.070	-0.001
SAE	0.436	-0.024	0.138	0.095	0.241	0.194	0.082	-0.000	0.091	0.033	-0.001	0.001	0.001	0.044	-0.001
							ΔP^-								
CGA_{pw}	0.088	**0.049**	0.096	0.047	0.036	0.021	0.056	0.017	**0.114**	0.071	-	-	-	-	-
CGA_{1s}	0.087	0.030	0.122	0.053	0.036	0.020	0.061	**0.028**	0.103	0.126	**0.010**	**0.159**	0.004	0.775	**0.006**
$\text{GRADIEND}_{\text{pw}}$	0.089	0.043	0.117	0.053	0.038	0.021	0.064	0.024	0.087	0.098	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.110	-0.044	**0.123**	**0.054**	**0.040**	**0.021**	**0.066**	0.021	0.109	**0.128**	0.009	0.153	-0.000	**0.849**	0.005
CAGA_{pw}	0.046	0.016	0.026	0.010	0.017	0.006	0.017	0.016	0.042	0.013	-	-	-	-	-
CAGA_{1s}	0.093	0.014	0.016	0.015	0.015	0.007	0.027	0.006	0.038	0.004	0.007	0.150	0.000	0.078	0.006
$\text{AGIEND}_{\text{pw}}$	0.133	0.042	0.093	0.032	0.015	0.013	0.059	0.005	0.052	0.073	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.135	0.028	0.110	0.032	0.004	0.018	0.061	0.009	0.044	0.106	-0.000	0.156	0.002	0.567	0.005
CAA_{pw}	0.139	0.021	0.119	0.051	0.035	0.021	0.063	0.010	0.104	0.087	-	-	-	-	-
CAA_{1s}	0.139	0.021	0.119	0.051	0.036	0.006	0.061	0.013	0.104	0.099	0.008	0.154	**0.005**	0.827	0.006
$\text{ACTIEND}_{\text{pw}}$	0.137	0.029	0.102	0.041	0.018	0.009	0.066	0.007	0.100	0.086	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.108	0.009	0.040	0.036	0.013	0.010	0.063	0.008	0.042	0.055	0.007	0.000	0.003	0.688	0.004
SAE	**0.139**	0.036	0.116	0.051	0.033	0.020	0.064	0.024	0.089	0.088	0.007	0.148	0.001	0.365	0.004
							Int.								
CGA_{pw}	0.143	0.067	0.085	0.056	0.057	0.071	0.050	0.008	**0.222**	0.088	-	-	-	-	-
CGA_{1s}	0.210	0.017	0.079	0.072	0.067	0.095	0.046	**0.014**	0.174	**0.224**	0.011	**0.143**	**0.157**	0.441	**0.396**
$\text{GRADIEND}_{\text{pw}}$	**0.294**	**0.107**	0.130	0.129	0.231	0.246	**0.273**	0.013	0.207	0.171	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.203	-0.021	**0.180**	**0.186**	**0.330**	**0.387**	0.252	0.010	0.114	0.110	**0.012**	0.077	0.000	**0.494**	0.003
CAGA_{pw}	0.025	0.009	0.014	0.005	0.010	0.004	0.009	0.008	0.024	0.009	-	-	-	-	-
CAGA_{1s}	0.047	0.007	0.009	0.009	0.009	0.003	0.021	0.003	0.023	0.003	0.004	0.083	0.013	0.041	0.003
$\text{AGIEND}_{\text{pw}}$	0.185	0.042	0.123	0.050	0.047	0.038	0.109	0.002	0.076	0.078	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.176	0.016	0.138	0.051	0.061	0.060	0.076	0.004	0.094	0.132	-0.000	0.092	0.043	0.351	0.003
CAA_{pw}	0.193	0.006	0.078	0.035	0.024	0.020	0.070	0.005	0.111	0.065	-	-	-	-	-
CAA_{1s}	0.193	0.006	0.085	0.050	0.030	0.041	0.052	0.006	0.111	0.082	0.004	0.076	0.002	0.428	0.003
$\text{ACTIEND}_{\text{pw}}$	0.213	0.025	0.075	0.047	0.016	0.013	0.080	0.004	0.111	0.068	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.147	0.005	0.055	0.042	0.013	0.023	0.094	0.004	0.061	0.053	0.005	0.000	0.001	0.379	0.002
SAE	0.287	0.006	0.127	0.073	0.137	0.107	0.073	0.012	0.090	0.061	0.003	0.074	0.001	0.205	0.002

Table 11: Detection raw results for GPT-2.

Method	Gender	Emotion	Race	Religion	RavelCont	RavelCntry	RavelLang	LangID	PronNum	PronPers	IOI	KeyValue	Induction	Repetition	FuncComp
							AUC_n								
$\mathrm{CGA}_{\mathrm{pw}}$	0.994	0.823	0.902	0.952	0.998	0.992	0.992	0.654	0.862	0.941	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.987	0.769	0.915	0.921	1.000	0.988	0.997	0.723	0.845	0.781	0.997	0.644	0.935	**1.000**	0.921
$\mathrm{GRADIEND}_{\mathrm{pw}}$	1.000	0.946	0.958	0.965	0.997	0.926	0.995	0.661	0.977	0.994	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	1.000	0.744	0.923	0.907	0.900	0.997	0.992	0.616	0.973	0.961	0.997	0.746	0.998	0.997	0.705
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.990	0.982	0.982	0.834	0.830	0.842	0.685	0.995	0.932	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.922	0.995	0.973	0.845	0.831	0.819	0.706	**1.000**	0.993	1.000	0.961	0.953	**1.000**	**1.000**
$\mathrm{AGIEND}_{\mathrm{pw}}$	**1.000**	**0.991**	0.989	0.998	0.998	0.970	0.989	0.790	0.993	0.997	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	**1.000**	0.692	0.972	0.981	0.995	0.954	0.964	0.768	0.992	0.993	**1.000**	0.486	0.982	**1.000**	0.796
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.849	**0.997**	0.999	**1.000**	1.000	1.000	**0.981**	1.000	1.000	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.849	0.997	**0.999**	**1.000**	1.000	**1.000**	0.980	1.000	**1.000**	1.000	**1.000**	0.957	**1.000**	1.000
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.954	0.883	0.916	0.932	0.922	0.920	0.909	0.816	0.920	0.914	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	0.887	0.751	0.934	0.929	0.827	0.881	0.960	0.782	0.900	0.930	0.648	0.498	0.612	0.966	0.768
SAE	1.000	0.501	0.953	0.916	1.000	**1.000**	1.000	0.941	0.869	0.978	1.000	1.000	**1.000**	**1.000**	0.998
							AUC_o								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.976	0.929	0.924	0.999	0.997	0.973	0.651	1.000	0.930	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.988	0.858	0.980	0.967	**1.000**	0.995	0.992	0.685	0.911	0.842	**1.000**	0.817	0.852	**1.000**	0.828
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.984	0.991	0.994	**1.000**	0.899	**1.000**	0.641	**1.000**	1.000	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	**1.000**	0.804	0.983	0.981	0.874	**1.000**	0.929	0.666	1.000	0.998	**1.000**	0.737	**0.940**	**1.000**	0.749
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.998	0.987	0.975	0.948	0.995	0.985	0.655	**1.000**	0.982	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.946	0.995	0.985	**1.000**	**1.000**	0.989	0.669	**1.000**	0.997	**1.000**	**0.853**	0.890	**1.000**	0.915
$\mathrm{AGIEND}_{\mathrm{pw}}$	**1.000**	0.999	1.000	**1.000**	**1.000**	0.972	**1.000**	0.854	**1.000**	1.000	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	**1.000**	0.872	0.989	0.999	**1.000**	0.935	0.979	0.854	1.000	1.000	**1.000**	0.618	0.908	**1.000**	0.905
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	**1.000**	0.967	0.991	**1.000**	**1.000**	**1.000**	**0.993**	**1.000**	1.000	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	**1.000**	**1.000**	0.999	**1.000**	**1.000**	**1.000**	0.992	**1.000**	**1.000**	**1.000**	0.757	0.837	**1.000**	**0.940**
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	**1.000**	0.996	0.998	**1.000**	**1.000**	**1.000**	0.937	**1.000**	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	**1.000**	0.998	0.976	0.996	**1.000**	**1.000**	**1.000**	0.837	**1.000**	**1.000**	0.997	0.550	0.636	**1.000**	0.671
SAE	1.000	0.500	0.950	0.916	**1.000**	**1.000**	**1.000**	0.934	0.869	0.978	0.908	0.645	0.631	**1.000**	0.834
							Spec_n								
$\mathrm{CGA}_{\mathrm{pw}}$	0.992	0.788	0.914	0.959	0.987	0.952	0.978	0.626	0.889	0.900	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.965	0.750	0.892	0.978	0.993	0.985	0.973	0.682	0.660	0.747	0.980	0.990	0.970	**1.000**	0.990
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.947	0.784	0.880	0.908	0.897	0.794	0.899	0.484	0.811	0.867	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	0.995	0.704	0.982	0.983	0.977	0.977	0.962	0.605	0.950	0.927	0.985	0.935	0.980	0.995	0.975
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.978	0.995	0.997	0.984	0.981	0.977	0.847	**1.000**	0.998	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.932	**1.000**	0.982	0.998	0.998	0.993	0.837	**1.000**	0.980	0.990	0.960	0.995	**1.000**	**1.000**
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.978	0.929	0.902	0.955	0.923	0.877	0.859	0.623	0.937	0.947	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	**1.000**	0.842	0.993	0.980	0.973	0.931	0.874	0.818	0.962	0.976	**1.000**	0.965	0.990	**1.000**	0.945
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	**0.986**	0.994	0.985	**1.000**	1.000	0.999	0.929	0.999	0.996	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	**0.986**	1.000	0.987	**1.000**	1.000	**1.000**	0.912	0.999	**0.999**	0.997	0.999	0.959	**1.000**	0.999
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.852	0.714	0.773	0.792	0.814	0.806	0.784	0.589	0.712	0.768	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	0.831	0.606	0.859	0.845	0.801	0.861	0.936	0.706	0.806	0.860	0.571	0.430	0.488	0.955	0.717
SAE	**1.000**	0.001	1.000	**1.000**	1.000	**1.000**	1.000	**0.976**	1.000	0.996	0.999	**0.999**	**0.996**	**1.000**	0.991
							Excl.								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.930	0.889	0.869	0.994	**1.000**	0.985	0.715	0.995	0.886	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.960	0.735	0.944	0.970	**1.000**	0.941	**1.000**	0.657	0.900	0.803	**1.000**	0.660	0.730	**1.000**	**0.960**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.970	0.939	0.974	**1.000**	0.964	**1.000**	0.631	**1.000**	0.997	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	**1.000**	0.686	0.922	0.882	0.822	**1.000**	0.964	0.604	0.985	0.987	**1.000**	0.610	0.802	**1.000**	0.895
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.990	0.987	0.982	0.932	**1.000**	0.992	0.649	**1.000**	0.990	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.870	0.995	0.980	**1.000**	**1.000**	0.970	0.768	**1.000**	0.995	**1.000**	0.720	0.770	**1.000**	0.880
$\mathrm{AGIEND}_{\mathrm{pw}}$	**1.000**	0.996	0.993	0.995	**1.000**	0.993	**1.000**	0.818	**1.000**	0.998	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	**1.000**	0.936	0.986	0.984	**1.000**	0.876	0.940	0.821	0.995	**1.000**	**1.000**	0.370	**0.830**	**1.000**	0.810
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	**1.000**	0.907	**1.000**	**1.000**	**1.000**	**1.000**	0.992	**1.000**	0.995	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	**1.000**	**1.000**	0.995	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**1.000**	**1.000**	0.610	0.810	**1.000**	0.810
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	**1.000**	0.993	0.995	**1.000**	**1.000**	**1.000**	0.886	**1.000**	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	**1.000**	0.977	0.964	0.997	**1.000**	**1.000**	**1.000**	0.784	**1.000**	**1.000**	0.995	0.185	0.380	**1.000**	0.305
SAE	**1.000**	**1.000**	0.996	0.998	**1.000**	**1.000**	**1.000**	0.968	**1.000**	0.994	0.838	**0.826**	0.552	**1.000**	0.563
							Det.								
$\mathrm{CGA}_{\mathrm{pw}}$	0.992	0.747	0.836	0.857	0.987	0.948	0.952	0.556	0.859	0.862	-	-	-	-	-
$\mathrm{CGA}_{\mathrm{1s}}$	0.959	0.708	0.890	0.921	0.993	0.941	0.973	0.588	0.660	0.708	0.980	0.644	0.730	**1.000**	0.828
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.947	0.772	0.849	0.877	0.896	0.764	0.899	0.374	0.811	0.867	-	-	-	-	-
$\mathrm{GRADIEND}_{\mathrm{1s}}$	0.995	0.639	0.920	0.880	0.821	0.977	0.908	0.523	0.950	0.927	0.985	0.610	0.802	0.995	0.705
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**0.972**	0.969	0.968	0.833	0.827	0.820	0.563	0.995	0.932	-	-	-	-	-
$\mathrm{CAGA}_{\mathrm{1s}}$	**1.000**	0.828	0.990	0.966	0.845	0.831	0.781	0.635	**1.000**	0.980	0.990	**0.720**	0.770	**1.000**	**0.880**
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.978	0.929	0.901	0.954	0.923	0.868	0.859	0.596	0.937	0.946	-	-	-	-	-
$\mathrm{AGIEND}_{\mathrm{1s}}$	**1.000**	0.638	0.972	0.971	0.973	0.836	0.868	0.748	0.962	0.976	**1.000**	0.370	**0.830**	**1.000**	0.796
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.843	0.905	0.978	**1.000**	1.000	0.999	**0.929**	0.999	0.991	-	-	-	-	-
$\mathrm{CAA}_{\mathrm{1s}}$	**1.000**	0.843	**0.997**	**0.984**	**1.000**	1.000	**1.000**	0.912	0.999	**0.999**	0.997	0.610	0.810	**1.000**	0.810
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.852	0.714	0.770	0.792	0.814	0.806	0.784	0.589	0.712	0.768	-	-	-	-	-
$\mathrm{ACTIEND}_{\mathrm{1s}}$	0.831	0.606	0.859	0.845	0.801	0.861	0.936	0.649	0.806	0.860	0.571	0.185	0.380	0.955	0.305
SAE	1.000	0.001	0.949	0.916	1.000	**1.000**	1.000	0.923	0.869	0.978	0.838	0.645	0.552	**1.000**	0.563

Table 12: Intervention raw results for GPT-2.

Method	GENDER	EMOTION	RACE	RELIGION	RAVELCONT	RAVELCNTRY	RAVELLANG	LANGID	PRONNUM	PRONPERS	IOI	KEYVALUE	INDUCTION	REPETITION	FUNCCOMP
							ΔP^+								
CGA_{pw}	0.343	0.016	0.142	0.214	0.383	0.414	0.388	0.000	0.201	-0.000	-	-	-	-	-
CGA_{1s}	0.270	0.003	0.090	0.049	**0.686**	**0.767**	**0.614**	0.000	0.121	0.111	0.179	0.263	0.314	**0.317**	**0.709**
$\text{GRADIEND}_{\text{pw}}$	**0.541**	0.043	0.136	0.158	0.535	0.330	0.479	**0.000**	0.298	**0.264**	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.120	0.002	0.015	0.352	0.570	0.253	0.574	0.000	0.293	0.260	0.327	0.190	**0.638**	0.309	0.009
CAGA_{pw}	0.376	0.005	0.182	0.162	0.403	0.211	0.544	0.000	0.157	0.165	-	-	-	-	-
CAGA_{1s}	0.406	0.003	0.260	0.294	0.499	0.509	0.529	0.000	0.077	0.200	**0.490**	**0.560**	0.574	0.307	0.475
$\text{AGIEND}_{\text{pw}}$	0.435	**0.139**	0.287	0.197	0.420	0.421	0.536	0.000	0.101	0.264	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.243	0.001	**0.498**	**0.462**	0.562	0.431	0.426	0.000	0.020	0.021	0.366	-0.036	0.592	0.309	0.590
CAA_{pw}	0.061	0.000	0.178	0.101	0.137	0.145	0.254	0.000	**0.300**	0.048	-	-	-	-	-
CAA_{1s}	0.061	0.000	0.205	0.052	0.138	0.167	0.246	0.000	**0.300**	0.077	0.025	0.012	0.020	0.151	0.015
$\text{ACTIEND}_{\text{pw}}$	0.247	0.014	0.080	0.035	0.106	0.147	0.195	0.000	0.116	0.045	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.242	0.000	0.096	0.038	0.090	0.110	0.271	-0.000	0.115	0.046	-0.000	-0.000	0.000	0.168	0.002
SAE	0.016	0.001	0.065	0.046	0.189	0.011	0.146	-0.000	0.121	0.026	0.123	0.185	0.122	0.058	0.055
							ΔP^-								
CGA_{pw}	0.267	0.035	0.093	0.088	0.075	0.100	0.102	0.002	0.120	0.024	-	-	-	-	-
CGA_{1s}	**0.268**	0.084	0.174	0.088	0.057	0.107	0.117	0.003	0.162	0.176	0.425	0.065	0.069	0.682	0.010
$\text{GRADIEND}_{\text{pw}}$	0.268	**0.120**	0.175	0.086	0.069	0.103	0.090	0.025	0.163	0.188	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.267	0.089	**0.175**	0.088	**0.075**	**0.107**	0.070	**0.047**	**0.163**	**0.191**	**0.425**	0.065	0.071	**0.686**	**0.010**
CAGA_{pw}	0.262	0.027	0.127	0.073	0.039	0.099	0.067	0.027	0.099	0.135	-	-	-	-	-
CAGA_{1s}	0.259	0.054	0.105	0.062	0.024	0.021	0.026	0.024	0.054	0.178	0.424	0.065	**0.072**	0.684	0.010
$\text{AGIEND}_{\text{pw}}$	0.267	0.084	0.121	0.060	0.045	0.061	0.088	0.003	0.079	0.180	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.265	0.066	0.078	0.046	0.023	0.042	0.067	0.005	0.144	0.189	0.424	0.064	0.070	0.684	0.009
CAA_{pw}	0.259	0.090	0.173	0.085	0.075	0.107	0.118	0.027	0.158	0.186	-	-	-	-	-
CAA_{1s}	0.259	0.090	0.173	0.085	0.075	0.107	0.117	0.025	0.158	0.187	0.424	0.064	0.071	0.685	0.010
$\text{ACTIEND}_{\text{pw}}$	0.263	0.080	0.173	**0.089**	0.063	0.107	0.094	0.001	0.138	0.096	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.262	0.066	0.159	0.084	0.050	0.107	0.114	-0.001	0.123	0.121	0.423	0.065	0.001	0.684	0.010
SAE	0.261	0.089	0.173	0.084	0.071	0.107	**0.118**	0.025	0.148	0.184	0.424	**0.065**	0.071	0.685	0.010
							Int.								
CGA_{pw}	0.305	0.026	0.118	0.151	0.229	0.257	0.245	0.001	0.161	0.012	-	-	-	-	-
CGA_{1s}	0.269	0.043	0.132	0.069	**0.371**	**0.437**	**0.366**	0.002	0.142	0.144	0.302	0.164	0.191	**0.500**	**0.360**
$\text{GRADIEND}_{\text{pw}}$	**0.404**	0.082	0.155	0.122	0.302	0.216	0.285	0.013	**0.231**	**0.226**	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.194	0.045	0.095	0.220	0.323	0.180	0.322	**0.024**	0.228	0.225	0.376	0.127	**0.355**	0.497	0.010
CAGA_{pw}	0.319	0.016	0.155	0.117	0.221	0.155	0.305	0.013	0.128	0.150	-	-	-	-	-
CAGA_{1s}	0.333	0.029	0.182	0.178	0.261	0.265	0.278	0.012	0.065	0.189	**0.457**	**0.312**	0.323	0.496	0.243
$\text{AGIEND}_{\text{pw}}$	0.351	**0.112**	0.204	0.129	0.232	0.241	0.312	0.001	0.090	0.222	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.254	0.033	**0.288**	**0.254**	0.293	0.237	0.246	0.002	0.082	0.105	0.395	0.014	0.331	0.497	0.300
CAA_{pw}	0.160	0.045	0.175	0.093	0.106	0.126	0.186	0.014	0.229	0.117	-	-	-	-	-
CAA_{1s}	0.160	0.045	0.189	0.068	0.106	0.137	0.182	0.012	0.229	0.132	0.224	0.038	0.046	0.418	0.013
$\text{ACTIEND}_{\text{pw}}$	0.255	0.047	0.127	0.062	0.085	0.127	0.145	0.001	0.127	0.071	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.252	0.033	0.127	0.061	0.070	0.109	0.192	-0.001	0.119	0.083	0.212	0.032	0.000	0.426	0.006
SAE	0.139	0.045	0.119	0.065	0.130	0.059	0.132	0.012	0.135	0.105	0.273	0.125	0.097	0.371	0.033

Table 13: Detection raw results for Llama-3.1-8B.

Method	Gender	Emotion	Race	Religion	RavelCont	RavelCntry	RavelLang	LangID	PronNum	PronPers	IOI	KeyValue	Induction	Repetition	FuncComp
							AUC_n								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.869	0.969	0.947	1.000	0.992	**1.000**	0.801	0.980	0.911	-	-	-	-	-
$\mathrm{CGA}_{1\mathrm{s}}$	**1.000**	0.813	0.945	0.925	0.997	0.939	1.000	0.952	0.996	0.967	**1.000**	1.000	**1.000**	**1.000**	0.998
$\mathrm{GRADIEND}_{\mathrm{pw}}$	1.000	0.625	0.797	0.935	0.997	0.931	0.990	0.573	0.891	0.984	-	-	-	-	-
$\mathrm{GRADIEND}_{1\mathrm{s}}$	0.999	0.652	0.903	0.872	0.991	0.878	0.941	0.579	0.846	0.910	1.000	0.729	0.954	**1.000**	0.617
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	0.930	0.965	0.990	1.000	0.851	0.900	0.680	0.990	0.932	-	-	-	-	-
$\mathrm{CAGA}_{1\mathrm{s}}$	0.999	0.811	0.985	0.994	1.000	0.870	0.919	0.686	1.000	0.983	**1.000**	**1.000**	0.943	1.000	0.989
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.927	0.903	0.891	0.868	0.968	0.843	0.932	0.648	0.926	0.941	-	-	-	-	-
$\mathrm{AGIEND}_{1\mathrm{s}}$	0.984	0.865	0.915	0.898	0.973	0.894	0.912	0.759	0.929	0.902	0.990	0.966	0.816	**1.000**	0.988
$\mathrm{CAA}_{\mathrm{pw}}$	0.996	0.759	0.999	0.996	1.000	**1.000**	**1.000**	0.994	**1.000**	1.000	-	-	-	-	-
$\mathrm{CAA}_{1\mathrm{s}}$	0.996	0.759	**0.999**	**0.999**	1.000	1.000	1.000	**0.994**	**1.000**	**1.000**	**1.000**	1.000	0.998	**1.000**	**1.000**
$\mathrm{ACTIEND}_{\mathrm{pw}}$	1.000	**0.990**	0.979	0.995	**1.000**	**1.000**	**1.000**	0.729	0.995	0.999	-	-	-	-	-
$\mathrm{ACTIEND}_{1\mathrm{s}}$	0.989	0.820	0.975	0.988	0.990	**1.000**	0.993	0.925	0.966	0.965	1.000	0.945	0.992	**1.000**	1.000
SAE	1.000	0.437	0.930	0.962	0.999	**1.000**	1.000	0.959	0.947	0.964	0.962	0.999	0.964	**1.000**	0.965
							AUC_o								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.950	0.949	0.899	0.987	0.965	0.996	0.748	0.996	0.889	-	-	-	-	-
$\mathrm{CGA}_{1\mathrm{s}}$	**1.000**	0.729	0.975	0.925	0.997	**1.000**	**1.000**	0.867	0.990	0.925	**1.000**	0.998	**0.985**	**1.000**	**1.000**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.661	0.842	0.976	**1.000**	0.972	0.995	0.555	0.936	0.999	-	-	-	-	-
$\mathrm{GRADIEND}_{1\mathrm{s}}$	**1.000**	0.668	0.828	0.958	0.999	0.888	0.969	0.542	0.899	0.950	1.000	0.816	0.867	**1.000**	0.537
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**1.000**	0.961	0.988	0.995	0.983	0.971	0.671	**1.000**	0.942	-	-	-	-	-
$\mathrm{CAGA}_{1\mathrm{s}}$	**1.000**	0.829	0.971	0.998	**1.000**	0.968	0.985	0.625	1.000	0.998	**1.000**	0.998	0.934	**1.000**	0.994
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.998	0.963	0.956	0.947	0.991	0.882	0.985	0.672	0.984	0.980	-	-	-	-	-
$\mathrm{AGIEND}_{1\mathrm{s}}$	0.991	0.817	0.916	0.916	0.939	0.984	0.956	0.792	0.948	0.908	**1.000**	0.875	0.783	**1.000**	0.500
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.896	0.998	0.997	**1.000**	**1.000**	**1.000**	0.995	**1.000**	0.996	-	-	-	-	-
$\mathrm{CAA}_{1\mathrm{s}}$	**1.000**	0.896	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.998**	**1.000**	**1.000**	**1.000**	**1.000**	0.934	**1.000**	1.000
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	1.000	0.997	1.000	**1.000**	**1.000**	**1.000**	0.761	1.000	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{1\mathrm{s}}$	0.997	0.979	0.997	0.999	**1.000**	**1.000**	**1.000**	0.965	1.000	1.000	**1.000**	0.773	0.975	**1.000**	0.999
SAE	1.000	0.486	0.937	0.973	**1.000**	**1.000**	**1.000**	0.958	0.992	0.997	1.000	0.999	0.944	**1.000**	0.749
							Spec_n								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.945	0.974	0.994	0.999	0.992	**1.000**	0.844	0.964	0.971	-	-	-	-	-
$\mathrm{CGA}_{1\mathrm{s}}$	**1.000**	0.953	0.958	0.973	0.993	0.988	0.995	0.960	0.991	0.926	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.952	0.502	0.754	0.928	0.920	0.859	0.888	0.349	0.729	0.868	-	-	-	-	-
$\mathrm{GRADIEND}_{1\mathrm{s}}$	0.995	0.467	0.948	0.978	0.950	0.888	0.955	0.398	0.822	0.831	**1.000**	0.985	0.925	**1.000**	0.745
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**0.992**	0.998	0.987	0.997	0.982	0.930	0.787	0.985	0.967	-	-	-	-	-
$\mathrm{CAGA}_{1\mathrm{s}}$	0.998	0.840	0.945	0.985	0.997	0.972	0.945	0.732	**1.000**	0.996	**1.000**	**1.000**	0.975	**1.000**	**1.000**
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.655	0.688	0.659	0.620	0.746	0.662	0.683	0.387	0.666	0.718	-	-	-	-	-
$\mathrm{AGIEND}_{1\mathrm{s}}$	0.968	0.778	0.898	0.865	0.947	0.883	0.913	0.648	0.877	0.823	0.980	0.940	0.950	**1.000**	0.975
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.986	1.000	**0.998**	1.000	**1.000**	**1.000**	**0.992**	0.999	**0.997**	-	-	-	-	-
$\mathrm{CAA}_{1\mathrm{s}}$	**1.000**	0.986	**1.000**	0.998	**1.000**	1.000	1.000	0.986	0.999	0.996	**1.000**	0.999	0.979	**1.000**	**1.000**
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.910	0.864	0.884	0.892	0.901	0.873	0.880	0.657	0.871	0.881	-	-	-	-	-
$\mathrm{ACTIEND}_{1\mathrm{s}}$	0.998	0.958	0.971	0.972	0.959	**1.000**	0.985	0.960	0.942	0.945	1.000	0.883	0.959	**1.000**	1.000
SAE	0.998	0.040	0.962	0.972	0.999	**1.000**	1.000	0.952	0.925	0.955	0.962	1.000	0.964	**1.000**	0.965
							Excl.								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.915	0.949	0.977	0.976	0.990	**1.000**	0.780	0.975	0.939	-	-	-	-	-
$\mathrm{CGA}_{1\mathrm{s}}$	**1.000**	0.720	0.949	0.934	0.982	**1.000**	**1.000**	0.864	0.990	0.869	**1.000**	0.970	0.900	**1.000**	0.990
$\mathrm{GRADIEND}_{\mathrm{pw}}$	**1.000**	0.619	0.834	0.906	**1.000**	0.930	0.995	0.586	0.880	0.991	-	-	-	-	-
$\mathrm{GRADIEND}_{1\mathrm{s}}$	**1.000**	0.548	0.801	0.905	0.985	0.804	0.970	0.333	0.835	0.844	0.995	0.740	0.675	**1.000**	0.665
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**1.000**	0.960	0.987	0.979	0.990	0.970	0.659	**1.000**	0.990	-	-	-	-	-
$\mathrm{CAGA}_{1\mathrm{s}}$	**1.000**	0.715	0.894	0.975	**1.000**	0.902	**1.000**	0.556	0.995	0.980	**1.000**	0.960	**0.940**	**1.000**	0.980
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.996	0.894	0.925	0.918	0.996	0.912	0.978	0.645	0.969	0.946	-	-	-	-	-
$\mathrm{AGIEND}_{1\mathrm{s}}$	0.982	0.742	0.824	0.861	0.879	0.959	0.940	0.831	0.905	0.854	**1.000**	0.755	0.795	**1.000**	**1.000**
$\mathrm{CAA}_{\mathrm{pw}}$	**1.000**	0.935	0.990	0.992	**1.000**	**1.000**	**1.000**	0.992	**1.000**	0.987	-	-	-	-	-
$\mathrm{CAA}_{1\mathrm{s}}$	**1.000**	0.935	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**1.000**	**1.000**	**1.000**	0.830	**1.000**	**1.000**
$\mathrm{ACTIEND}_{\mathrm{pw}}$	**1.000**	0.989	0.977	0.996	**1.000**	**1.000**	**1.000**	0.643	0.996	**1.000**	-	-	-	-	-
$\mathrm{ACTIEND}_{1\mathrm{s}}$	0.985	0.943	0.990	0.997	**1.000**	**1.000**	**1.000**	0.903	0.992	0.994	**1.000**	0.750	0.865	**1.000**	0.995
SAE	0.998	0.965	0.996	0.995	**1.000**	**1.000**	**1.000**	0.962	0.997	0.993	0.999	0.995	0.841	**1.000**	0.530
							Det.								
$\mathrm{CGA}_{\mathrm{pw}}$	**1.000**	0.811	0.931	0.893	0.975	0.957	0.996	0.726	0.964	0.879	-	-	-	-	-
$\mathrm{CGA}_{1\mathrm{s}}$	**1.000**	0.576	0.934	0.897	0.980	0.938	0.995	0.857	0.987	0.863	**1.000**	0.970	0.900	**1.000**	0.990
$\mathrm{GRADIEND}_{\mathrm{pw}}$	0.952	0.422	0.657	0.866	0.920	0.818	0.888	0.286	0.720	0.867	-	-	-	-	-
$\mathrm{GRADIEND}_{1\mathrm{s}}$	0.995	0.467	0.791	0.859	0.942	0.793	0.936	0.215	0.791	0.787	0.995	0.729	0.675	**1.000**	0.537
$\mathrm{CAGA}_{\mathrm{pw}}$	**1.000**	**0.930**	0.930	0.975	0.978	0.832	0.877	0.580	0.985	0.917	-	-	-	-	-
$\mathrm{CAGA}_{1\mathrm{s}}$	0.998	0.573	0.874	0.972	0.997	0.773	0.904	0.496	0.995	0.978	**1.000**	0.960	**0.934**	1.000	0.980
$\mathrm{AGIEND}_{\mathrm{pw}}$	0.655	0.684	0.652	0.620	0.746	0.644	0.680	0.365	0.666	0.713	-	-	-	-	-
$\mathrm{AGIEND}_{1\mathrm{s}}$	0.960	0.708	0.824	0.842	0.874	0.879	0.884	0.637	0.876	0.792	0.980	0.755	0.783	**1.000**	0.500
$\mathrm{CAA}_{\mathrm{pw}}$	0.996	0.722	0.990	0.991	1.000	**1.000**	**1.000**	**0.986**	**0.999**	0.985	-	-	-	-	-
$\mathrm{CAA}_{1\mathrm{s}}$	0.996	0.722	**0.999**	**0.998**	**1.000**	1.000	1.000	0.983	**0.999**	**0.996**	**1.000**	**0.999**	0.830	**1.000**	**1.000**
$\mathrm{ACTIEND}_{\mathrm{pw}}$	0.910	0.864	0.867	0.889	0.901	0.873	0.880	0.504	0.866	0.880	-	-	-	-	-
$\mathrm{ACTIEND}_{1\mathrm{s}}$	0.977	0.820	0.959	0.972	0.959	**1.000**	0.985	0.891	0.942	0.945	1.000	0.750	0.865	**1.000**	0.995
SAE	0.998	0.040	0.914	0.955	0.999	**1.000**	1.000	0.949	0.925	0.954	0.962	0.995	0.841	**1.000**	0.530

Table 14: Intervention raw results for Llama-3.1-8B.

Method	Gender	Emotion	Race	Religion	RavelCont	RavelCntry	RavelLang	LangID	PronNum	PronPers	IOI	KeyValue	Induction	Repetition	FuncComp
							ΔP^+								
CGA_{pw}	0.265	**0.084**	0.095	0.098	0.595	0.433	0.325	-0.000	0.396	0.232	-	-	-	-	-
CGA_{1s}	0.500	0.001	0.002	0.089	0.385	0.131	0.101	0.000	0.294	0.076	0.397	**0.810**	**0.589**	**0.170**	**0.406**
$\text{GRADIEND}_{\text{pw}}$	**0.595**	0.001	0.338	0.162	**0.704**	**0.799**	**0.645**	-0.000	0.270	**0.580**	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.449	-0.000	0.017	0.004	0.544	0.464	0.611	0.000	**0.404**	0.288	0.111	0.679	0.456	0.170	0.086
CAGA_{pw}	0.013	-0.002	0.003	0.001	0.018	0.002	0.027	-0.000	0.003	0.017	-	-	-	-	-
CAGA_{1s}	0.080	-0.001	0.001	0.003	0.016	0.001	0.029	-0.000	0.009	0.005	**0.555**	-0.003	0.040	0.017	0.074
$\text{AGIEND}_{\text{pw}}$	0.334	0.062	**0.488**	**0.386**	0.600	0.455	0.510	0.004	0.272	0.318	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.166	0.001	0.278	0.339	0.563	0.581	0.327	0.000	0.238	0.151	0.437	0.241	0.116	0.023	0.093
CAA_{pw}	0.001	0.000	0.179	0.057	0.125	0.002	0.018	-0.000	0.364	0.152	-	-	-	-	-
CAA_{1s}	0.001	0.000	0.246	0.191	0.099	0.057	0.312	-0.000	0.364	0.106	0.060	0.660	0.368	0.028	0.116
$\text{ACTIEND}_{\text{pw}}$	0.332	0.024	0.026	0.032	0.463	0.297	0.256	**0.013**	0.099	0.055	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.332	0.000	0.024	0.051	0.193	0.006	0.220	0.000	0.123	0.049	0.019	0.017	0.187	0.035	0.047
SAE	0.213	-0.002	0.017	0.080	0.177	0.001	0.194	0.002	0.056	0.013	0.025	0.001	0.011	-0.001	0.001
							ΔP^-								
CGA_{pw}	0.410	0.149	0.326	0.161	0.805	0.668	0.588	0.208	0.116	0.222	-	-	-	-	-
CGA_{1s}	0.410	0.166	0.341	0.249	0.805	0.692	0.588	0.212	0.244	0.323	**0.328**	0.151	0.342	0.830	0.051
$\text{GRADIEND}_{\text{pw}}$	0.410	0.209	0.355	0.182	**0.806**	0.691	0.548	0.213	0.251	0.327	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.410	0.175	0.355	0.181	0.806	0.691	0.588	**0.214**	**0.263**	**0.328**	0.328	**0.151**	0.341	**0.830**	**0.051**
CAGA_{pw}	0.409	0.170	0.357	0.260	0.804	0.692	0.589	0.213	0.256	0.323	-	-	-	-	-
CAGA_{1s}	0.409	0.166	**0.357**	0.260	0.804	0.692	**0.589**	0.213	0.253	0.320	0.328	0.150	**0.342**	0.808	0.051
$\text{AGIEND}_{\text{pw}}$	**0.410**	0.209	0.355	0.267	0.806	**0.692**	0.587	0.213	0.263	0.324	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.410	0.175	0.355	0.265	0.806	**0.692**	0.587	0.213	0.263	0.325	0.328	0.099	0.341	0.814	0.050
CAA_{pw}	0.409	0.175	0.357	0.260	0.804	0.691	0.587	0.212	0.256	0.323	-	-	-	-	-
CAA_{1s}	0.409	0.175	0.357	0.260	0.804	0.691	0.587	0.212	0.256	0.323	0.328	0.150	0.342	0.829	0.050
$\text{ACTIEND}_{\text{pw}}$	0.410	**0.211**	0.354	**0.268**	0.694	0.425	0.393	0.183	0.243	0.326	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.407	0.090	0.344	0.267	0.806	0.691	0.413	0.213	0.235	0.293	0.328	0.151	0.341	0.830	0.051
SAE	0.408	0.172	0.356	0.260	0.804	**0.692**	**0.589**	0.212	0.250	0.262	0.328	0.134	0.342	0.829	0.047
							Int.								
CGA_{pw}	0.337	0.117	0.210	0.130	0.700	0.551	0.456	0.104	0.256	0.227	-	-	-	-	-
CGA_{1s}	0.455	0.083	0.172	0.169	0.595	0.411	0.345	0.106	0.269	0.199	0.363	**0.481**	**0.465**	**0.500**	**0.229**
$\text{GRADIEND}_{\text{pw}}$	**0.502**	0.105	0.346	0.172	**0.755**	**0.745**	0.596	0.106	0.260	**0.454**	-	-	-	-	-
$\text{GRADIEND}_{\text{1s}}$	0.430	0.087	0.186	0.093	0.675	0.578	**0.599**	0.107	**0.333**	0.308	0.219	0.415	0.399	**0.500**	0.068
CAGA_{pw}	0.211	0.084	0.180	0.130	0.411	0.347	0.308	0.106	0.129	0.170	-	-	-	-	-
CAGA_{1s}	0.244	0.083	0.179	0.131	0.410	0.346	0.309	0.106	0.131	0.162	**0.441**	0.074	0.191	0.413	0.062
$\text{AGIEND}_{\text{pw}}$	0.372	**0.136**	**0.422**	**0.327**	0.703	0.574	0.549	**0.108**	0.268	0.321	-	-	-	-	-
$\text{AGIEND}_{\text{1s}}$	0.288	0.088	0.316	0.302	0.684	0.636	0.457	0.106	0.251	0.238	0.383	0.170	0.228	0.419	0.072
CAA_{pw}	0.205	0.088	0.268	0.159	0.465	0.346	0.303	0.106	0.310	0.237	-	-	-	-	-
CAA_{1s}	0.205	0.088	0.302	0.226	0.452	0.374	0.450	0.106	0.310	0.214	0.194	0.405	0.355	0.429	0.083
$\text{ACTIEND}_{\text{pw}}$	0.371	0.117	0.190	0.150	0.579	0.361	0.325	0.098	0.171	0.190	-	-	-	-	-
$\text{ACTIEND}_{\text{1s}}$	0.370	0.045	0.184	0.159	0.499	0.349	0.316	0.106	0.179	0.171	0.173	0.084	0.264	0.432	0.049
SAE	0.310	0.085	0.187	0.170	0.491	0.346	0.392	0.107	0.153	0.137	0.177	0.067	0.176	0.414	0.024