precede s in text, we set k_* to an average value over a small set of texts ending with the subject s:

$$k_* = \frac{1}{N}\sum_{j=1}^{N} k(x_j + s), \text{ where } k(x) = \sigma\left(W_{fc}^{(l^*)}\;\gamma(a_{[x],i}^{(l^*)} + h_{[x],i}^{(l^*-1)})\right). \tag{3}$$

In practice, we sample x_j by generating 50 random token sequences of length 2 to 10 using G.

Step 2: Choosing v_* to Recall the Fact. Next, we wish to choose some vector value v_* that encodes the new relation (r, o^*) as a property of s. We set $v_* = \operatorname{argmin}_z \mathcal{L}(z)$, where the objective $\mathcal{L}(z)$ is:

$$\frac{1}{N}\sum_{j=1}^{N} \underbrace{-\log \mathbb{P}_{G(m_i^{(l^*)}:=z)}\left[o^* \mid x_j + p\right]}_{\text{(a) Maximizing } o^* \text{ probability}} + \underbrace{D_{\mathrm{KL}}\left(\mathbb{P}_{G(m_{i'}^{(l^*)}:=z)}\left[x \mid p'\right] \big\| \mathbb{P}_G\left[x \mid p'\right]\right)}_{\text{(b) Controlling essence drift}}. \tag{4}$$

The first term (Eqn. 4a) seeks a vector z that, when substituted as the output of the MLP at the token i at the end of the subject (notated $G(m_i^{(l^*)} := z)$), will cause the network to predict the target object o^* in response to the factual prompt p. The second term (Eqn. 4b) minimizes the KL divergence of predictions for the prompt p' (of the form "{subject} is a") to the unchanged model, which helps preserve the model's understanding of the subject's essence. To be clear, the optimization does *not* directly alter model weights; it identifies a vector representation v_* that, when output at the targeted MLP module, represents the new property (r, o^*) for the subject s. Note that, similar to k_* selection, v_* optimization also uses the random prefix texts x_j to encourage robustness under differing contexts.

Step 3: Inserting the Fact. Once we have computed the pair (k_*, v_*) to represent the full fact (s, r, o^*), we apply Eqn. 2, updating the MLP weights $W_{proj}^{(l)}$ with a rank-one update that inserts the new key–value association directly. For full implementation details, see Appendix E.5.

3.2 Evaluating ROME: Zero-Shot Relation Extraction (zsRE)

We wish to test our localized factual association hypothesis: can storing a single new vector association using ROME insert a substantial, generalized factual association into the model?

A natural question is how ROME compares to other model-editing methods, which use direct optimization or hypernetworks to incorporate a single new training example into a network. For baselines, we examine Fine-Tuning **(FT)**, which applies Adam with early stopping at one layer to minimize $-\log \mathbb{P}\left[o^* \mid x\right]$. Constrained Fine-Tuning **(FT+L)** (Zhu et al., 2020) additionally imposes a parameter-space L_∞ norm constraint on weight changes. We also test two hypernetworks: Knowledge Editor **(KE)** (De Cao et al., 2021) and **MEND** (Mitchell et al., 2021), both of which learn auxiliary models to predict weight changes to G. Further details are described in Appendix E.

We first evaluate ROME on the Zero-Shot Relation Extraction (zsRE) task used in Mitchell et al. (2021) and De Cao et al. (2021). Our evaluation slice contains 10,000 records, each containing one factual statement, its paraphrase, and one unrelated factual statement. "Efficacy" and "Paraphrase" measure post-edit accuracy $\mathbb{I}\big[o^* = \operatorname{argmax}_o \mathbb{P}_{G'}\left[o\right]\big]$ of the statement and its paraphrase, respectively, while "Specificity" measures the edited model's accuracy on an unrelated fact. Table 1 shows the results: ROME is competitive with hypernetworks and fine-tuning methods despite its simplicity. We find that it is not hard for ROME to insert an association that can be regurgitated by the model. Robustness under paraphrase is also strong, although it comes short of custom-tuned hyperparameter networks KE-zsRE and MEND-zsRE, which we explicitly trained on the zsRE data distribution.[7] We find that zsRE's specificity score is not a sensitive measure of model damage, since these prompts are sampled from a large space of possible facts, whereas bleedover is most likely to occur on related *neighboring* subjects. Appendix C has additional experimental details.

Table 1: zsRE Editing Results on GPT-2 XL.

Editor	**Efficacy** ↑	**Paraphrase** ↑	**Specificity** ↑
GPT-2 XL	22.2 (±0.5)	21.3 (±0.5)	24.2 (±0.5)
FT	99.6 (±0.1)	82.1 (±0.6)	23.2 (±0.5)
FT+L	92.3 (±0.4)	**47.2 (±0.7)**	23.4 (±0.5)
KE	65.5 (±0.6)	61.4 (±0.6)	24.9 (±0.5)
KE-zsRE	92.4 (±0.3)	90.0 (±0.3)	23.8 (±0.5)
MEND	75.9 (±0.5)	65.3 (±0.6)	24.1 (±0.5)
MEND-zsRE	99.4 (±0.1)	**99.3 (±0.1)**	24.1 (±0.5)
ROME	**99.8 (±0.0)**	88.1 (±0.5)	**24.2 (±0.5)**

[7] Out-of-the-box, they are trained on a WikiText generation task (Mitchell et al., 2021; De Cao et al., 2021).