Algorithm 1: Attention Suppression Loss

Input: Attention matrix $\mathbf{A}_{\text{r2all}}$; Index sets $\mathcal{I}_p$ and $\mathcal{I}_{np}$
Parameter: Threshold factor γ; top-k value k
Output: Suppression loss $\mathcal{L}_{\text{sup}}$

Obtain $\mathbf{A}_{\text{prompt}}$ and $\mathbf{A}_{\text{non-prompt}}$ from $\mathbf{A}_{\text{r2all}}$ using index sets $\mathcal{I}_p$ and $\mathcal{I}_{np}$
$\mathbf{T}_k \leftarrow \text{TopK}(\mathbf{A}_{\text{prompt}},\ k)$
$\mu_{\text{non}} \leftarrow \text{Mean}(\mathbf{A}_{\text{non-prompt}})$
return $\mathcal{L}_{\text{sup}} = \text{Mean}(\text{ReLU}(\mathbf{T}_k / \mu_{\text{non}} - \gamma))$

This loss can be implemented in a single line of Python:

```
loss = ReLU(TopK(A_prompt, k) / Mean(A_non_prompt) - factor).mean()
```