A Proofs of Theorems

We now present the proofs of the various theorems presented in the main manuscript. We recall the definitions presented in Section 5. The data can be represented as query-item pairs, given by $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{z})$. A key design choice is on forming the dataset $\mathbb{S} = \{\boldsymbol{s}\}$. The simplest formulation is that of selecting N queries and items randomly from the underlying distributions $\mathcal{X}$ and $\mathcal{Z}$ respectively, and pairing them, to form the dataset $\tilde{\mathbb{S}} = \{\boldsymbol{s}_i = (\boldsymbol{x}_i, \boldsymbol{z}_i) \mid i = 1, 2, \ldots N\}$. Alternatively, given $\mathbb{X}$ and $\mathbb{Z}$, defined as $\mathbb{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, and $\mathbb{Z}_s = \{\boldsymbol{z}_\ell\}_{\ell=1}^{L}$ respectively, we can consider all possible pairing of queries and items, giving us $\mathbb{S} = \{(\boldsymbol{x}_i, \boldsymbol{z}_\ell) \mid i = 1, 2, \ldots N, \ell = 1, 2, \ldots L\}$, *i.e.,* $\boldsymbol{s}$ is drawn from the Cartesian product space $\mathcal{S} = \mathcal{X} \times \mathcal{Z}$. Note that $\mathbb{S}$ can be indexed as $j = L(i-1) + \ell,\ j = 1, 2, \ldots M = NL$. The extreme (meta) classification problem is now one of binary classification, with targets y drawn from the space $\mathcal{Y}$, that satisfy $y_j = 1$ if item $\boldsymbol{z}_\ell$ is positively associated with the query $\boldsymbol{x}_i$. In the XC setting, it is well known that negatively associated pairs are significantly more likely to occur than positive ones. We assume that in any set $\mathbb{S}$, we have at most κ positively associated pairs. Let $\max_{\boldsymbol{x} \in \mathcal{E}(\mathcal{X})} \|\boldsymbol{x}\|_2 \leq B$ and $\max_{\boldsymbol{w} \in \mathbb{W}} \|\boldsymbol{w}\|_2 \leq W$ be the bounds on the norm of the encoder representations of the queries and the learnt classifiers, respectively.

Before we proceed with the proofs, we recall a few key definitions.

Empirical and True Risk: Given a function $f(\cdot)$ drawn from a function space $\mathcal{F}$, and a loss function, the empirical risk over $\mathbb{S}$, and the true risk, are given by

$$\hat{R} = \frac{1}{M} \sum_{\substack{j=1 \\ \boldsymbol{s}_j \sim \mathbb{S}}}^{M} \text{loss}(f(\boldsymbol{s}_j), y_j) \quad \text{and} \quad R = \mathbb{E}_{\boldsymbol{s} \sim \mathcal{X} \times \mathcal{Z}} \left[\text{loss}(f(\boldsymbol{s}), y)\right], \quad \text{respectively.}$$

Rademacher Complexity [32]: Given a function f drawn from the function class $\mathcal{F}$, the emprical Rademacher complexity defined over the set $\mathbb{S}$, and the Rademacher complexity over all sets of size M are given by:

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) = \frac{1}{M} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j f(\boldsymbol{s}_j) \right], \quad \text{and} \quad \mathfrak{R}_M(\mathcal{F}) = \mathbb{E}_{\mathbb{S}} \left[\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \right], \tag{8}$$

respectively, where $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_M)$, whose entries are independent random variables drawn from the *Rademacher distribution* i.e. $\Pr(\sigma_j = +1) = \Pr(\sigma_j = -1) = 1/2,\ \forall j$.

McDiarmid Inequality [11]: McDiarmid's inequality is a concentration inequality to bound the deviation of the sampled value from the expected value. We consider an extension of the standard inequality, to the case where the function under consideration Φ does not strictly satisfy the bounded differences property, but large differences remain very rare.

Let $\Phi : \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M \to \mathbb{R}$ be a function acting on the dataset $\mathbb{S}$, with $\boldsymbol{s}_j \in \mathcal{S}_j$ and $\mathcal{S}' \subseteq \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_M$ be a subset of its domain and let $c_1, c_2, \ldots, c_n \geq 0$ be constants such that all pairs $(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) \in \mathcal{S}'$ and $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_n) \in \mathcal{S}'$, satisfy the bounded difference property:

$$\left|\Phi(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_M) - \Phi(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_M)\right| \leq \sum_{j : \boldsymbol{s}_j \neq \boldsymbol{s}'_j} c_j.$$

All datasets drawn from $\mathcal{S}'$ can be viewed as *good sets* that satisfy bounded difference. Then, given a random dataset $\mathbb{S}$, with probability $q = 1 - \Pr(\mathbb{S} \in \mathcal{S}')$ (which is the probability of drawing a *bad set*), the following holds:

$$\Pr\left(|\Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right]| \geq \epsilon\right) \leq 2q + 2\exp\left\{-\frac{2\left(\max\left\{0, \epsilon - q\sum_j c_j\right\}\right)^2}{\sum_j c_j^2}\right\}.$$

Hoeffding's Inequality [32]: The Hoeffding's inequality is a special case of the McDiarmid inequality. Let $Y_1, Y_2, \ldots Y_M$ be M independent Bernoulli-distributed random variables. Let the sum be denoted as $S_M = Y_1 + Y_2 + \ldots + Y_M$. Then, $\forall t > 0$, we have

$$\Pr\left(S_M - \mathbb{E}_Y\left[S_M\right] \geq t\right) \leq \exp\left\{-\frac{2t^2}{M}\right\}.$$

Proof of Theorem 1: Consider the empirical and true risk defined in Equation 8. The proof follows by deriving the generalization bound using the McDiarmid inequality, followed by using the Hoeffding inequality to bound the probability q. Without loss of generality, we redefine the loss to have the target labels $y_j \in \{-1, 1\}$, giving rise to the following form of the loss:

$$g(\boldsymbol{s}, y) = \text{loss}\left(f(\boldsymbol{s}), y\right) = \left(\frac{1-y}{2}\right) f(\boldsymbol{s}) - C\left(\frac{1+y}{2}\right) f(\boldsymbol{s}). \tag{9}$$

Let $\Phi(\mathbb{S})$ be defined as follows

$$\Phi(\mathbb{S}) = \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}_{\boldsymbol{s}}\left[g(\boldsymbol{s}, y)\right] - \frac{1}{M}\sum_{j=1}^{M} g(\boldsymbol{s}_j, y_j) \right\} = \sup_{f \in \mathcal{F}} \left\{ \mathbb{E}\left[g(\boldsymbol{s}, y)\right] - \hat{\mathbb{E}}\left[g(\boldsymbol{s}_j, y_j)\right] \right\}$$

The subsequent steps are consistent with those in deriving the Rademacher complexity in the standard setting [32], but modified to accommodate the extension of the McDiarmid inequality. Let $\mathbb{S}$ and $\mathbb{S}'$ be two sample sets, differing only in element $\boldsymbol{s}_k$. Then, we have:

$$\Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] \leq \frac{1}{M} \sup_{f \in \mathcal{F}} \left\{ g(\boldsymbol{s}_k, y_k) - g'(\boldsymbol{s}_k, y_k) \right\} \leq c_j$$

if $\mathcal{S}$ and $\mathcal{S}'$ are drawn from the *good set* $\tilde{\mathcal{S}}$ (which we discuss shortly), then, we have $c_j = \frac{1}{M}$, $\forall j$, and McDiarmid's inequality can be applied to $\Phi(\mathbb{S})$ to obtain the following, for $\epsilon > q$:

$$\begin{aligned}
\Pr\left(\left|\Phi(\mathbb{S}) - \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right]\right| \geq \epsilon\right) &\leq 2q + 2\exp\left\{-\frac{2\left(\max\left\{0, \epsilon - q\sum_j c_j\right\}\right)^2}{\sum_j c_j^2}\right\} \\
&= 2q + 2\exp\left\{-2M\left(\max\left\{0, \epsilon - q\right\}\right)^2\right\} \\
&= \underbrace{2q + 2\exp\left\{-2M\left(\epsilon - q\right)^2\right\}}_{\delta}
\end{aligned}$$

Solving for ϵ on the right hand side, we obtain:

$$\begin{aligned}
&\delta = 2q + 2\exp\left\{-2M\left(\epsilon - q\right)^2\right\} \\
\Rightarrow\ &0 = \epsilon^2 + (-2q)\epsilon + \left(q^2 + \frac{1}{2M}\ln\left(\frac{2}{\delta - 2q}\right)\right) \\
\Rightarrow\ &\epsilon = q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}},
\end{aligned}$$

for $\delta \in (2q, 1)$. Substituting the above into McDiarmid's inequality, we obtain, with probability $1 - \delta$,

$$\Phi(\mathbb{S}) \leq \mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] + \left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right).$$

It is straightforward to derive the bound $\mathbb{E}_{\mathbb{S}}\left[\Phi(\mathbb{S})\right] \leq 2\mathfrak{R}_M(\text{loss} \circ \mathcal{F})$ [32]. Similarly, by applying the McDiarmid inequality to $\mathfrak{R}$, we get:

$$\mathfrak{R}_M(\text{loss} \circ \mathcal{F}) \leq \hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) + \left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right).$$

Further, for the given form of the loss, we have the following result:

$$\begin{aligned}
\hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) &= \frac{1}{M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j (\text{loss} \circ f)(\boldsymbol{s}_j)\right] \\
&= \frac{1}{M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j \left(\left(\frac{1 - y_j}{2}\right) f(\boldsymbol{s}_j) - C\left(\frac{1 + y_j}{2}\right) f(\boldsymbol{s}_j)\right)\right] \\
&= \frac{1}{2M}\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j f(\boldsymbol{s}_j)(1 - C) - \sigma_j y_j f(\boldsymbol{s}_j)(1 + C)\right]
\end{aligned}$$

Since σ_j and $-\sigma_j y_j$ are distributed identically, we have:

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\text{loss} \circ \mathcal{F}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{\sigma}}\left[2\frac{1}{M}\sup_{f \in \mathcal{F}} \sum_{j=1}^{M} \sigma_j f(\boldsymbol{s}_j)\right] = \hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F})$$

Putting the above together, and substituting in for $\Phi(\mathbb{S})$, we obtain

$$R \leq \hat{R} + \hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) + 3\left(q + \sqrt{\frac{\ln\left(\frac{2}{\delta - 2q}\right)}{2M}}\right),$$

which completes part of the proof of Theorem 1.

What remains, is to derive q, the probability of drawing a *bad set* $\mathbb{S}$. A *good set* is one which satisfies the bounded difference condition. Consider the set $\mathbb{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_M\}$. Let p denote the probability that a query-item pair is positively correlated. Then, by the definition of the loss in Equation 9, for positively correlated pairs, the maximum error (over all $f \in \mathcal{F}$ is W, while for negatively correlated pairs, it is 1. For $c_j = \frac{1}{M}$, we require that the sample $\boldsymbol{s}_j$ is a negative pair. Let a *good set* $\mathbb{S}$ have at most κ positively associated pairs. Then, a *bad set* contains at least $M - \kappa$ positively associated pairs. Then, q is the probability of drawing a set $\mathbb{S}$ with at least $M - \kappa$ positively associated pairs.

To derive this probability, consider indicator variables $Y_j = \mathbb{I}_{[\boldsymbol{s}_j \text{ contains a positive pairing}]}$. Let $S_M = \sum_j Y_j$ denote the sum. Then, we have $\mathbb{E}_Y[S_M] = pM$. Applying the Hoeffding inequality with $t = M - \kappa - pM$, we get:

$$\Pr\left(S_M - \mathbb{E}_Y[S_M] \geq t\right) \leq 2\exp\left\{-\frac{2t^2}{M}\right\}$$

$$\Rightarrow \Pr\left(S_M - pM \geq M - \kappa - pM\right) = \Pr\left(S_M \geq M - \kappa\right) \leq 2\exp\left\{-\frac{2(M - \kappa - pM)^2}{M}\right\}.$$

Simplifying, we get:

$$q = \Pr(\text{set } \mathbb{S} \text{ contains at least } M - \kappa \text{ positively associated pairs}) \leq 2\exp\left\{-2M\left(1 - p - \frac{\kappa}{M}\right)^2\right\},$$

which is significantly smaller than 1, given large M (which is true in the case of XC, where we have large N and L) and $\kappa = 0$, which is the setting under which the required McDiarmid inequality is defined. This completes the proof of Theorem 1.

Proof of Lemma 4: We now state and discuss the proof of Lemma 4, which is an extension of Theorem 3 from Awasthi et al. [4].

Lemma. *(**Rademacher complexity of the XC Classifiers**) (extension of Awasthi et al. [4], Theorem 3) Let $\mathcal{F}$ be the class of linear classifiers defined over the seen-item set $\mathbb{Z}_s$ in the classical XC setting (cf. Section 3.1), i.e., $\mathcal{F} = \{\langle \boldsymbol{x}, \boldsymbol{w}_\ell \rangle \mid \ell = 1, 2, \ldots, L\}$, where $\boldsymbol{x} \in \mathbb{X}$. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \leq \frac{LBW}{\sqrt{N}},$$

where $|\mathbb{X}| = N$ is the Cardinality of the training set.

Proof. Let $\mathcal{F}_\ell = \{\langle \boldsymbol{x}, \boldsymbol{w}_\ell \rangle \mid \|\boldsymbol{w}_\ell\|_2 \leq W\}$. Awasthi et al. [4] showed that $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{W}{N}\|\mathbf{X}\|_F$, where $\mathbf{X}$ is a matrix formed with the elements of $\mathbb{X}$. For bounded data, $\max_{\boldsymbol{x} \in \mathcal{E}(\mathcal{X})} \|\boldsymbol{x}\|_2 \leq B$, we have $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{WB}{\sqrt{N}}$. Given $\mathcal{F}$ with L classifiers, the Rademacher complexity is bounded by $\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}_\ell) \leq \frac{LBW}{\sqrt{N}}$. □

Proof of Lemma 2 and Corollary 3: We now derive the bound on the Rademacher complexity of the IRENE meta-classifier generator. Recall the Lemma

Lemma. *(**Rademacher complexity of the IRENE generator**) Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm, comprising pre-determined encoder representations and classifiers, a given classifier selector that outputs K classifiers, and $\mathcal{G}$, the meta-classifier generator. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \leq O\left(B\|\mathbf{M}\|_2\sqrt{d\ln(K+1)}\right),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times 1}$ is the weight matrix associated with the linear layer.

Proof. Recall the IRENE meta-classifier generator:

$$\mathcal{F} = \left\{(\boldsymbol{x}_i, z_\ell) \to \left\langle \boldsymbol{x}_i, \text{Linear}\left(\text{SelfAttention}\left(C_{lf}(\mathcal{S}(z_\ell))\right)\right)\right\rangle\right\}, \quad \text{where}$$
$$\mathcal{F}_3 = C_{lf}(\mathcal{S}(z_\ell)),\ \mathcal{F}_2 = \text{SelfAttention}(\mathcal{F}_3),\ \text{and } \mathcal{F}_1 = \text{Linear}(\mathcal{F}_2) = \mathbf{M}\mathcal{F}_2 + \boldsymbol{b}$$

The proof follows by repeatedly applying Talagrand's lemma, which states that, given an L-Lipschitz continuous function g, and the function class $\mathcal{F}$, we have $\hat{\mathfrak{R}}_{\mathbb{S}}(g \circ \mathcal{F}) \leq L\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F})$. Applying Talagrand's lemma to $\mathcal{F}$, we get:

$$\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) \leq B\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_1) \tag{10}$$
$$\leq B\|\mathbf{M}\|_2\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_2) \tag{11}$$
$$\leq B\|\mathbf{M}\|_2\sqrt{d\ln(K+1)}\,\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_3) \tag{12}$$
$$\leq O\left(B\|\mathbf{M}\|_2\sqrt{d\ln(K+1)}\right), \tag{13}$$

where Equation 10 is a consequence of the boundedness of $\boldsymbol{x}$, Equation 11 is obtained by considering the Lipschitz constant of a linear transformation layer with vector-valued weights, and Equation 12 is obtained by applying the Lipschitz constant of a self-attention layer,

derived by Vuckovic et al. [43], where in turn, d is the dimensionality of the input sequence, and K is the number of classifiers selected by $\mathcal{S}$. We note that, while Kim et al. [27] provide a tighter bound in the context of L_2 multi-head attention, their result does not hold for the dot-product-based self-attention block considered above. Equation 13 is a consequence of considering a fixed set of classifiers $\mathbb{W}$, and a pre-determined classifier selector algorithm. Therefore, these blocks merely add a constant factor to the complexity to the model $\mathcal{F}$ class, completing the proof of Lemma 2. □

The analysis can be extended to derive Corollary 3 by incorporating trainable classifiers. Recall the statement of the Corollary:

Corollary. *(**Rademacher complexity of the IRENE generator with trainable classifier**) Let $\mathcal{F}$ be the class of functions defined in the IRENE algorithm as in Lemma 2. Let the classifier set $\mathbb{W}$ be trainable over the meta-classifier loss. Then, the Rademacher complexity of $\mathcal{F}$ can be bounded as follows:*

$$\hat{\mathfrak{R}}_{\mathbb{S}}(\mathcal{F}) \leq O\left(B^2 W \sqrt{\frac{L}{M}} \|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\right),$$

where $\boldsymbol{x} \in \mathbb{R}^d$ and $\mathbf{M} \in \mathbb{R}^{d \times 1}$ is the weight matrix associated with the linear layer.

Proof. The above result can be obtained by extending the proof of Lemma 2:

$$\begin{aligned}\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) &\leq B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\, \hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_3) \\ &\leq O\left(B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\, \hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_{\text{clf}})\right),\end{aligned}$$

where $\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}_{\text{clf}})$ can be obtained from Lemma 4, considering the dataset $\mathbb{S}$ with $M = NL$ samples, and L classifiers associated with the observed items, which yields the desired result:

$$\begin{aligned}\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{F}) &\leq O\left(B\|\mathbf{M}\|_2 \sqrt{d \ln(K+1)} \frac{LW}{M} \|\mathbf{X}\|_F\right) \\ &= O\left(\sqrt{\frac{L}{N}} B^2 W \|\mathbf{M}\|_2 \sqrt{d \ln(K+1)}\right).\end{aligned}$$

This completes the proof of Corollary 3. □

B Dataset Creation and Statistics

We evaluate IRENE on a diverse set of datasets spanning multiple applications, such as product recommendation (LF-AmazonTitles-1.3M), category annotation (LF-Wikipedia-500K), query completion (LF-AOL-270K), and taxonomy completion (LF-WikiHierarchy-600K). Additionally, we also evaluate IRENE on a proprietary query-to-keyword matching dataset. Table 7 describes the statistics of these datasets.

Zero-shot splits of these datasets were created in the following manner: Given the original dataset consisting of L items, and a split fraction $s \in (0, 1)$, sL items were randomly selected to form the novel item set. The remaining $(1 - s)L$ items, along with their associated queries, form the training corpus. The novel test set was created based on connections between test queries and novel items, while for the generalized evaluation, we refer to the data source from [9] and [7]. We fix the notation *dataset-split* (e.g., LF-AmazonTitles-1.3M-10 for a 10% unseen ratio split) to refer to the zero-shot version of the dataset.

Table 7: Statistics of different datasets used to benchmark IRENE

Dataset	No. of Queries			No. of Items	
	Observed	Novel	Gen	Observed	Novel
KeywordPrediction-10M	220,845,427	5,022,052	5,022,052	4,999,996	5,435,443
LF-AOL-270K-10	3,689,542	68,491	519,352	245,543	27,282
LF-WikiHierarchy-550K-10	1,587,567	339,086	397,870	494,733	54,970
LF-AmazonTitles-1.3M-10	2,225,354	624,830	970,237	1,174,739	130,526
LF-Wikipedia-500K-10	1,781,890	271,620	783,743	450,963	50,107

C IRENE Implementation Details

For the classifier selector $\mathcal{S}$, based on the ablations, we set the number of neighbors K to be 3, and the number of transformer layers D to be 1. The selector is a transformer layer with 4 heads, and is trained with a dropout value of 0.1. The latent dimension is 768. We train the models with a batch size of 2048, and a learning rate of 0.0002 for 40 epochs. The margin is set to 0.2 on LF-Wikipedia-500K-10, 0.1 on LF-WikiHierarchy-550K-10, and 0.3 on both LF-AmazonTitles-1.3M-10 and LF-AOL-270K-10. The source code, with additional implementation details for IRENE, is available at https://aka.ms/irene.

Table 8: Ablation study on meta-classifier generator $\mathcal{G}$ and classifier selector $\mathcal{S}$ component in NGAME + IRENE on LF-WikiHierarchy-550K-10, LF-AOL-270K-10, LF-AmazonTitles-1.3M-10, and LF-Wikipedia-500K-10 datasets for zero-shot evaluation. The depth D denotes the number of layers in the transformer-based meta-classifier generator $\mathcal{G}$, while K denotes the numbers of neighbors selected by $\mathcal{S}$. We observe that smaller values for $K \in \{2, 3\}$ and $D \in \{1, 2\}$ yield superior results. Setting $D = 1$ and $K = 3$ works reasonably well for diverse datasets and base encoders.

	Ablation	LF-AOL-270K-10			LF-WikiHierarchy-550K-10			LF-AmazonTitles-1.3M-10			LF-Wikipedia-500K-10		
		P@1	P@5	R@10	P@1	P@5	R@10	P@1	P@5	R@10	P@1	P@5	R@10
	IRENE ($D = 1$, $K = 3$)	36.47	11.46	59.57	69.29	38.81	80.40	31.56	16.57	38.83	44.91	15.60	67.79
$\mathcal{G}$	$D = 2$, $K = 3$	35.05	11.11	57.88	70.36	39.06	80.39	31.65	16.55	38.83	48.46	16.20	69.49
	$D = 4$, $K = 3$	36.56	11.48	59.51	70.71	39.11	80.27	31.42	16.44	38.61	48.47	16.10	69.14
Selector ($\mathcal{S}$)	$D = 1$, $K = 1$	36.02	11.28	58.53	68.98	38.56	80.11	31.24	16.38	38.45	45.87	15.83	68.59
	$D = 1$, $K = 2$	36.54	11.42	59.35	69.34	38.74	80.25	31.47	16.52	38.75	45.45	15.74	68.19
	$D = 1$, $K = 6$	35.66	11.30	58.95	69.99	39.12	80.79	31.68	16.65	39.17	45.24	15.44	67.23
	$D = 1$, $K = 20$	36.15	11.36	59.05	69.07	38.57	79.80	31.50	16.59	39.14	45.75	15.30	66.89

D Ablations Discussions

Meta-classifier Generator $\mathcal{G}$: In the context of generating a new item representation $\mathbf{u}_\ell$, let $\mathcal{S}(\mathbf{z}_\ell)$ denote the set of shortlisted seen items for a given item l, with $\mathbf{w}_\ell \in \mathcal{S}(\mathbf{z}_\ell)$. Here, $\mathbf{w}_\ell^i$ represents the classifier of the i^{th} neighbor, and $\mathbf{z}_\ell$ is the encoder representation of the new item.

Summation Formulation: The summation formulation for $\mathbf{u}_\ell$ is given by aggregating the encoder representation of the new item with the classifiers of its K neighbors as follows:

$$\mathbf{u}_\ell = \left(\mathbf{z}_\ell + \mathbf{w}_\ell^1 + \mathbf{w}_\ell^2 + \ldots + \mathbf{w}_\ell^K\right) \tag{14}$$

Weighted Summation Formulation: In the weighted summation formulation, $\mathbf{u}_\ell$ is computed by a weighted sum of the encoder representation and the classifiers of its neighbors, where the weights c^i are learned parameters:

$$\mathbf{u}_\ell = \left(c^0\mathbf{z}_\ell + c^1\mathbf{w}_\ell^1 + c^2\mathbf{w}_\ell^2 + \ldots + c^K\mathbf{w}_\ell^K\right) \tag{15}$$

Classifier Selector $\mathcal{S}$: We discuss detailed ablations on the classifier selector. First, we evaluate the effect of changing K, the number of observed items retrieved by $\mathcal{S}$, given an ANNS-based $\mathcal{S}$, on zero-shot performance. From the results presented in Table 4, we observe that increasing K from 3 to 6 results in a performance decline of approximately 3% in P@1. Further, increasing K to 20 continues to decrease performance. This is consistent with observations made in Section 5, wherein smaller K yield a tighter generalization bound, as derived in Lemma 2.

Second, to demonstrate the flexibility of the IRENE framework in diverse applications wherein latency is not critical,as a proof of concept, we consider a classifier selector comprising a GPT-4-based re-ranking model that re-ranks the neighbour shortlist obtained from encoder embeddings. This ablation is carried out on a subset of 10^4 novel items. We observe P@1 and R@10 improvements by 1%, while the representation time increased by $O(10^4)$ (cf. Table 9). Given an item, the representation time is the time for re-ranking neighbours and subsequently, generating representations using $\mathcal{G}$.

Table 9: Proof-of-concept experiments on the zero-shot performance comparison of the ANNS-based and GPT-4-based classifier selectors in IRENE. When experimented on a small, but random, subset of the data, employing the GPT-4-based re-ranking model results in 1% improvements in terms of precision of recall but the time for re-ranking neighbours and generating the representations (Rep. time) increases by an order of four.

Classifier Selector ($\mathcal{S}$)	P@1 ↑	R@10 ↑	Rep. Time (ms) ↓
ANNS-based $\mathcal{S}$	67.14	85.77	0.43
GPT-4-based $\mathcal{S}$	68.26	86.61	4000

E Detailed Discussion on Sponsored Search

Sponsored search is essentially a match-making system between users and advertisers with the objective of optimizing user experience, while searching for knowledge. Sponsored search enables advertisers to reach the right set of users who might be interested in their product/service [1]. Users encode their intent in short pieces of text called queries. Similarly, advertisers bid on short pieces of text, relevant

Table 10: Results on 100M zero-shot keywords on the search engine. Method Ms are anonymized in-production dense retrievers. All algorithms are provided with just the text of a keyword to get its representation

Method	R@30	R@50	R@100	P@30	P@50	P@100
M1	24.74	32.82	48.46	39.10	36.60	33.25
M2	19.93	26.49	38.99	35.72	33.11	29.63
M3	19.11	25.84	39.33	31.48	28.93	25.69
M4	28.79	39.64	62.19	44.24	41.96	38.72
NGAME+IRENE	**30.53**	**42.00**	**66.68**	**45.64**	**43.40**	**40.11**

Table 11: Expert judges labeling results on KeywordPrediction-10M dataset

Method	% of good quality predictions
NGAME	64.06
NGAME+IRENE	73.16
NGAME+IRENE- OneShot	77.05

to their ads, known as keywords. One critical component of the sponsored search pipeline is the task of matching user queries to these advertiser bid keywords. Most search engines currently follow varying semantics to perform this matching (called match-types [8]). Matching user queries to advertiser keywords is a nuanced and challenging problem as maintaining the semantics of the match-type is essential to advertisers who often bid differently on different match-types for the same keyword[36]. Furthermore, given the placement of this matching task at the forefront of the Ads retrieval pipeline, any improvements in accuracy within this application can yield super-linear benefits for downstream components [45]. We note that the choice of a larger truncation factor (30, 50, and 100) for the Precision metric is deliberate – Retrieval algorithms such as IRENE precede re-ranking algorithms, necessitating the prediction of a larger candidate set for input into these re-ranking algorithms.

Online Results IRENE underwent deployment on a prominent search engine for conducting A/B tests with live search-engine traffic. Throughout the live A/B test interaction on the search engine, IRENE was systematically compared against an extensive control ensemble featuring diverse algorithms, encompassing not only DR algorithms but also prominent generative, graph-based, and IR algorithms. Performance evaluation was based on live metrics. The findings revealed that IRENE led to a 4.2% increase in the click-through rate (ad clicks obtained per unit query) and a 0.9% decrease in the quick-back rate (fraction of users quickly leaving the ad landing page due to perceived irrelevance). These results underscore the value creation for users, indicating that IRENE effectively presented more relevant ads to the audience. Furthermore, IRENE demonstrated a noteworthy 7.8% increase in keyword density (average number of keywords surviving quality control and relevance filters), affirming the quality of its predictions. Additionally, IRENE achieved a click efficiency of 150%, signifying that for every 2% increase in ad impressions, ads selected by IRENE garnered 3% more clicks. When labeled by expert judges, IRENE was found to increase the percentage of *good keyword* predictions by 9% (refer to Table 11 for details). Notably, IRENE successfully matched queries such as "grainger" and "bitwarden" to advertiser keywords like "industrial supply" and "password manager," respectively. It is essential to highlight that these predictions, not relying on text matching, were not replicated by any in-production algorithm. Please refer to Table 12 for more such predictions made by IRENE but missed by the control ensemble.

Furthermore, we conduct a direct comparison of IRENE with prominent proprietary and public Dense Retrieval (DR) algorithms currently in production. Specifically, we randomly sample 100 million advertiser keywords introduced into the system after the period covered by the training-data scraping. Additionally, we select some of the top-performing dense retrieval encoders deployed in production and pit IRENE against them in recommending keywords from this 100-million set for a sampled array of queries. For intellectual property reasons, the names of these algorithms are anonymized, and the results are detailed in Table 10. IRENE was found to be at least 4% superior to the next best dense retriever in terms of R@100. As novel items stream into the system, it is necessary to frequently encode these items and include them in the ANNS index. Table 3 in the Main Manuscript shows that IRENE adds only minimal overhead on top of a language encoder and can get the item representation in less than one millisecond. Further, the integration of updatable ANNS algorithms, such as Fresh-DiskANN [40], can greatly reduce deployment time for novel items.

Offline Results To conduct offline experiments, we curated the KeywordPrediction-10M dataset by mining the logs of a commercial search engine within a specific timeframe. The dataset comprised user-typed queries and the corresponding bid keywords for surfaced advertisements, forming query-keyword training pairs. These pairs underwent basic sanity filters based on click-through rate (CTR), clicks, and impressions to generate the training dataset. Named KeywordPrediction-10M, the dataset encompassed approximately 5 million items and 220 million training queries. For additional details, refer to table 7. As presented in Table 13, IRENE demonstrated a superiority of at least 3% in Recall@100 and at least 2% in Precision@30 compared to leading Dense Retrieval (DR) algorithms NGAME and ANCE.

Table 12: Advertiser keywords predicted by IRENE for a user query, but overlooked by the production ensemble comprising leading dense retrieval, graph-based, XC, and generative language models. IRENE extends its capability beyond textual similarity by endowing the query representation with world knowledge obtained from classifiers of similar observed items

User Query	Advertiser Keyword
best pfmea control plan software	best cmms
work order app hydraulics	plumbing work order software
firewalla gold att fiber connection	att ethernet network
nn2 best home fumigation companies ventura oxnard	bug removal in oxnard
youtube	streaming services
netbenefits com login	fidelity retirement account
financial services crm software	sap customer management software
hypokalemia	high potassium in the blood
kaiser options	healthinsurance
23andme	genetic screening

Table 13: Results on KeywordPrediction-10M dataset.

Method	R@30	R@50	R@100	P@30	P@50	P@100
Evaluation only on novel items						
ANCE	31.37	42.87	72.45	76.3	68.40	56.32
NGAME	33.76	48.07	74.13	76.52	69.32	57.67
NGAME+IRENE	**34.79**	**49.81**	**77.32**	**78.70**	**71.76**	**60.18**
Semsup-XC	34.43	49.06	74.99	77.53	70.15	58.04
NGAME+IRENE-OneShot	**35.87**	**50.95**	**78.46**	**79.75**	**72.84**	**61.20**

One-shot extension We further study the extension of IRENE to scenarios involving items that have received precisely one click, representing an exploration of IRENE's performance at the extreme tail of observed items. In this context, IRENE leverages the revealed query for a one-shot tail item to enhance its classifier selector. The revealed query of the one-shot item is utilized to retrieve the nearest classifiers, along with the item itself. A max-voting strategy is then employed to select superior observed classifiers compared to cases where only the item text is used for this purpose. In comparison to SemsupXC, which refines its language encoder with new click data, IRENE demonstrated a superiority of approximately 3% and 2% in Recall@100 and Precision@30, respectively. It's noteworthy that fine-tuning models deployed in production, as undertaken by SemsupXC, introduces latency and complexity costs, making it less desirable. Additionally, algorithms that fine-tune the trained model on revealed data, such as SemsupXC, necessitate rebuilding the Approximate Nearest Neighbors (ANNS) index from scratch. In contrast, IRENE adopts updatable ANNS algorithms, akin to many Dense Retrieval (DR) algorithms, allowing it to leverage revealed data for improved prediction accuracy without the need to fine-tune the base model. Hence, IRENE can make use of the revealed data to improve the prediction accuracy without having to fine-tune the base model. This helps to reduce latency and complexity in an online serving infrastructure.

Table 14: Zero Shot Accuracies of different encoders when combined with IRENE. Averaged across base encoders and datasets, IRENE improves P@1, P@5, and R@10 by 9%, 4.2%, and 10.1%, respectively.

Model	LF-AOL-270K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	43.90	48.80	54.20	60.14	65.39	30.90	15.43	10.32
NGAME+IRENE	49.59	54.40	59.57	65.78	71.12	36.47	17.39	11.46
ANCE	51.81	59.63	67.84	77.38	84.78	33.43	18.10	12.52
ANCE+IRENE	53.44	60.29	67.82	76.47	83.23	36.84	18.67	12.66
MACLR	13.97	15.63	18.24	21.46	30.73	11.31	4.96	3.33
MACLR+IRENE	48.15	54.20	61.29	70.54	78.86	34.32	16.89	11.41
DPR	43.44	48.53	53.82	59.71	64.84	30.38	15.26	10.24
DPR+IRENE	50.14	55.13	60.22	66.18	71.49	36.80	17.57	11.61
TF-IDF	20.10	23.76	28.05	34.10	39.13	13.74	7.27	5.08
Zest-XML	22.73	23.49	25.91	29.28	35.1	9.34	10.21	10.79
Adam	30.49	33.96	38.53	45.81	56.31	23.02	10.80	7.22
SemSup-XC	31.50	33.8	36.31	39.93	42.92	26.27	11.16	7.19
DEXA	31.64	36.50	41.85	48.81	55.59	21.68	11.16	7.73

Model	LF-WikiHierarchy-550K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	37.96	47.24	58.66	73.67	84.31	46.01	32.93	25.63
NGAME+IRENE	59.02	70.42	80.40	88.83	93.85	69.29	50.98	38.81
ANCE	35.97	44.92	56.28	72.30	85.56	43.06	30.68	23.98
ANCE+IRENE	59.35	71.50	82.10	90.70	95.29	66.54	50.52	38.87
MACLR	21.97	27.33	35.47	44.33	66.89	30.37	19.07	14.44
MACLR+IRENE	60.11	71.76	81.43	89.95	95.02	69.45	51.76	39.41
DPR	37.14	47.34	59.29	74.6	85.64	44.84	32.3	25.53
DPR+IRENE	59.42	70.45	80.01	88.72	93.91	69.65	51.23	38.78
TF-IDF	14.46	17.19	21.44	29.57	39.14	22.79	12.53	9.08
Zest-XML	14.56	16.39	17.48	29.28	22.85	13.97	13.29	12.67
Adam	29.31	36.08	45.04	58.39	71.76	38.3	25.19	19.13
SemSup-XC	40.56	44.39	46.81	48.95	50.09	57.45	36.11	24.67
DEXA	45.96	55.68	66.89	79.3	87.47	54.83	39.59	30.29

Model	LF-AmazonTitles-1.3M-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	24.20	29.40	36.44	46.71	56.04	30.42	19.94	15.38
NGAME+IRENE	25.36	31.14	38.83	49.86	59.36	31.56	21.28	16.57
ANCE	18.14	23.32	30.72	43.66	57.76	22.38	15.14	12.02
ANCE+IRENE	19.55	24.93	32.72	45.48	58.56	22.75	16.21	13.02
MACLR	17.59	21.96	28.59	40.35	53.38	21.93	14.50	11.39
MACLR+IRENE	17.77	22.31	28.77	39.18	49.89	21.56	14.82	11.71
DPR	25.72	32.07	40.98	53.87	64.18	31.10	21.29	16.82
DPR+IRENE	25.40	31.60	40.31	52.63	62.18	30.49	21.04	16.62
TF-IDF	8.12	10.18	13.30	18.92	25.63	24.15	18.31	15.04
Zest-XML	5.42	6.35	6.87	7.89	8.67	5.58	4.71	4.22
SemSup-XC	11.28	14.73	20.04	29.26	38.27	11.68	8.41	6.85
DEXA	23.17	28.27	35.19	45.29	54.27	28.83	18.98	14.66

Model	LF-Wikipedia-500K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	53.55	58.91	65.27	74.33	82.07	46.96	22.56	15.10
NGAME+IRENE	54.19	60.59	67.79	76.63	83.52	44.91	22.90	15.60
ANCE	40.46	47.71	58.91	75.70	87.98	30.67	16.57	11.92
ANCE+IRENE	54.84	62.59	71.59	82.91	91.16	41.59	23.02	16.05
MACLR	51.02	58.91	68.53	81.3	91.7	39.56	21.37	15.05
MACLR+IRENE	56.38	64.13	73.05	83.63	91.39	44.64	23.76	16.48
DPR	55.28	62.70	71.20	81.22	89.43	42.90	23.02	15.96
DPR+IRENE	54.55	62.08	70.50	80.45	88.03	42.19	22.83	15.87
TF-IDF	15.43	18.64	23.89	34.66	48.24	11.53	6.29	4.60
Zest-XML	6.86	9.95	14.73	21.63	25.55	2.62	2.62	2.32
SemSup-XC	50.38	53.80	57.08	60.01	61.03	46.60	21.46	13.90
DEXA	52.45	59.15	67.37	78.04	87.36	42.76	21.91	15.07

Table 15: Generalized Zero-Shot Accuracies of different encoders when combined with IRENE. Averaged across base encoders and datasets, IRENE improves P@1, P@5, and R@10 by 14.9%, 10.4%, and 9.8%, respectively.

Model	LF-AOL-270K-10							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	24.71	30.70	38.27	47.93	55.24	20.16	13.81	10.43
NGAME+IRENE	39.31	45.15	52.30	61.42	68.87	35.11	20.43	14.43
ANCE	31.06	39.29	49.72	58.74	74.00	22.63	15.86	12.25
ANCE+IRENE	36.93	43.53	51.75	63.04	72.89	30.84	18.67	13.78
MACLR	5.87	6.55	7.52	8.62	12.36	9.26	4.22	2.83
MACLR+IRENE	32.30	37.62	44.71	55.32	65.59	30.40	17.16	12.25
DPR	24.35	30.37	37.99	44.50	54.95	19.71	13.61	10.31
DPR+IRENE	39.51	45.43	52.57	61.73	69.13	35.07	20.54	14.52
TF-IDF	5.81	7.40	9.90	14.26	19.88	6.61	4.14	3.16
Zest-XML	19.67	22.18	26.57	29.8	33.39	26.34	14.71	10.19
SemSup-XC	20.98	22.55	23.92	25.21	25.86	26.12	13.97	9.08
DEXA	31.50	38.51	46.76	52.98	62.96	25.09	16.35	12.35
Model	**LF-WikiHierarchy-550K-10**							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	11.24	16.70	27.08	39.85	68.14	66.19	62.64	59.25
NGAME+IRENE	15.25	23.32	40.09	72.71	87.81	91.33	89.52	86.67
ANCE	11.02	16.36	25.89	37.67	65.11	68.76	63.48	58.87
ANCE+IRENE	15.06	23.06	39.74	72.87	89.43	90.72	88.77	85.97
MACLR	7.23	9.89	14.31	19.63	36.31	59.44	47.20	39.93
MACLR+IRENE	14.48	22.33	38.37	70.92	87.89	88.81	87.19	84.60
DPR	11.08	16.45	26.73	39.47	68.76	65.19	61.53	58.14
DPR+IRENE	14.94	23.01	39.84	72.74	88.02	89.52	87.91	85.56
TF-IDF	6.67	8.36	10.88	15.56	22.16	64.50	42.24	32.1
Zest-XML	11.76	16.87	22.13	26.68	29.83	68.86	49.19	38.73
SemSup-XC	13.80	19.93	28.37	32.15	32.38	90.51	84.87	78.61
DEXA	13.74	21.12	36.17	53.96	81.81	76.18	76.94	75.38
Model	**LF-AmazonTitles-1.3M-10**							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	17.45	22.58	30.25	42.72	54.81	45.14	39.15	34.72
NGAME+IRENE	17.69	23.17	31.49	45.09	58.19	47.77	42.68	38.35
ANCE	9.45	12.41	17.31	27.25	40.62	27.65	22.76	19.76
ANCE+IRENE	11.00	15.25	22.34	35.85	51.26	36.78	32.41	29.31
MACLR	9.13	11.76	15.99	24.57	36.57	27.50	21.86	18.60
MACLR+IRENE	9.57	13.05	18.85	30.13	43.61	31.49	26.92	23.97
DPR	14.85	19.48	26.93	35.42	55.12	38.18	32.89	29.20
DPR+IRENE	15.38	20.73	29.25	44.28	59.42	43.08	38.66	35.00
TF-IDF	13.71	16.47	20.40	27.08	34.81	16.33	9.83	7.35
Zest-XML	12.34	14.45	22.87	26.79	39.71	41.36	33.7	28.29
SemSup-XC	7.76	10.59	15.21	23.41	30.87	25.13	20.93	18.37
DEXA	18.08	23.27	30.89	38.70	54.21	48.19	40.45	35.47
Model	**LF-Wikipedia-500K-10**							
	R@3	R@5	R@10	R@30	R@100	P@1	P@3	P@5
NGAME	52.24	60.96	69.58	78.67	85.50	81.86	60.13	45.38
NGAME+IRENE	50.29	59.56	69.27	79.78	87.27	78.99	58.60	44.86
ANCE	29.66	35.51	43.39	56.22	71.99	42.91	27.54	20.92
ANCE+IRENE	43.91	52.88	63.46	76.42	86.32	71.39	49.56	38.09
MACLR	29.20	36.38	46.62	61.86	75.69	46.59	31.12	24.36
MACLR+IRENE	42.80	51.95	62.82	75.93	85.77	70.52	51.10	39.24
DPR	38.93	49.21	61.84	72.17	87.01	51.54	40.30	32.71
DPR+IRENE	44.88	54.95	66.71	80.17	89.10	70.39	52.50	40.91
TF-IDF	9.49	11.67	14.79	20.87	30.10	15.07	9.19	6.93
Zest-XML	32.24	38.01	45.15	54.31	59.32	60.16	39.33	29.21
SemSup-XC	23.44	28.83	38.08	48.01	61.30	54.20	40.72	29.58
DEXA	45.83	55.05	65.86	74.63	88.11	67.98	48.88	37.90

Table 16: Ablation study on meta-classifier generator $\mathcal{G}$ and classifier selector $\mathcal{S}$ component in IRENE on LF-WikiHierarchy-550K-10 dataset. Here IRENE is NGAME+IRENE

Method	Zero shot								Generalized							
	R@3	**R@5**	**R@10**	**R@30**	**R@100**	**P@1**	**P@3**	**P@5**	**R@3**	**R@5**	**R@10**	**R@30**	**R@100**	**P@1**	**P@3**	**P@5**
IRENE ($\mathcal{D}=1, K=3$)	59.02	70.42	80.40	88.83	93.85	69.29	50.98	38.81	15.25	23.32	40.09	72.71	87.81	91.33	89.52	86.67
$\mathcal{D}=2$	59.83	70.83	80.39	88.60	93.79	70.36	51.66	39.06	15.35	23.49	40.48	73.51	88.35	92.13	90.10	87.24
$\mathcal{D}=4$	60.00	70.85	80.27	88.51	93.70	70.71	51.82	39.11	15.37	23.52	40.56	73.72	88.42	92.28	90.18	87.33
$\mathcal{G}$ as Sum	37.55	47.10	59.01	73.87	84.80	45.49	32.59	25.46	11.23	16.84	27.15	39.57	56.20	68.11	64.01	60.51
$\mathcal{G}$ as wt. Sum	38.34	47.72	59.29	74.33	85.12	46.39	33.23	25.88	11.29	16.80	27.28	40.25	57.74	66.60	63.08	59.72
IRENE + Enc. Embed.	49.10	60.60	72.36	82.31	87.66	56.79	42.14	32.92	14.25	21.49	35.89	53.75	73.33	87.17	83.12	79.29
$K=1$	58.82	70.24	80.11	88.79	93.84	68.98	50.79	38.56	15.13	23.13	39.67	72.01	87.59	91.04	88.85	85.95
$K=2$	59.18	70.33	80.25	88.94	93.94	69.34	51.01	38.74	15.23	23.27	39.97	72.41	87.74	91.15	89.30	86.40
$K=6$	59.63	70.83	80.79	88.94	93.81	69.94	51.52	39.12	15.09	23.14	39.93	72.98	88.04	89.93	88.59	86.01
$K=20$	58.67	69.81	79.80	88.06	93.30	69.07	50.78	38.57	15.23	23.27	39.93	72.75	87.92	91.76	89.61	86.64