WHAT IF RECOMMENDER SYSTEMS WERE CIRCUITS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommender system research largely focused on learning richer user and item representations and designing novel objective functions, while many approaches still model user-item interactions as a simple factorization realized via a dot product. We revisit this factorised component from a probabilistic perspective, asking whether it can support tractable probabilistic inference and learning, without redesigning the underlying recommender architecture. To this end, we introduce *CiRe*, a plug-in interaction head that replaces the usual dot product with a nonnegative squared circuit, thus delivering a tractable generative model that can be trained with maximum likelihood, avoiding the use of negative sampling and supporting probabilistic queries over user–item interactions. We show that across nine recommender architectures and five datasets, CiRe frequently matches or improves top-k ranking performance with the strongest results combining circuit parametrisation and likelihood training. For simple encoders, removing negative sampling can also reduce training time, whereas richer architectures exhibit different computational trade-offs, which we analyse. Our code is available at https://anonymous.4open.science/r/CiRe-2B56/.

1 INTRODUCTION

Recommender systems (RSs) aim to identify the items that are most relevant to a user from observed interactions (Jannach et al., 2010). A large class of recommenders inherits the basic structure of matrix factorisation: users and items are mapped to latent representations and combined through a simple factorised interaction, typically a dot product (Koren et al., 2009). Recent work has focused on improving how these representations are constructed, through graph propagation (He et al., 2020), self-supervision (Wu et al., 2021), and multimodal information (Attimonelli et al., 2026), or on developing richer learning objectives to improve recommendation accuracy (Zhou et al., 2023; Zhou & Shen, 2023), often introducing additional training components and computational cost. Many factorisation-based recommenders are trained with Bayesian Personalized Ranking (BPR) (Rendle et al., 2009), sometimes alongside auxiliary objectives. Such objectives rely on negative sampling to construct pairwise preference comparisons, introducing additional training overhead and making the learning signal dependent on the sampling strategy (Rendle & Freudenthaler, 2014). Yet, many of these models still combine their final user and item representations through the same factorised interaction (Malitesta et al., 2025).

In this work, we revisit this shared factorised interaction from a probabilistic perspective. Recent results connecting tensor factorisations and *probabilistic circuits* (PCs) (Darwiche, 2003; Vergari et al., 2019; Choi et al., 2020) provide the theoretical basis for this direction. PCs are structured computational graphs composed of sums and products that, under suitable structural conditions, support tractable operations such as exact normalisation and marginalisation (Vergari et al., 2021). Low-rank tensor factorisations admit equivalent sum–product circuit representations, placing matrix factorisation—the interaction underlying many recommenders—within this computational framework (Loconte et al., 2025a). Previous work has further shown that factorised predictors can be transformed into nonnegative, tractably normalisable probability models through squared circuit constructions (Loconte et al., 2023). Overall, these results suggest a principled route for turning the factorised interaction of existing recommenders into a tractable probabilistic model, enabling probabilistic queries over user–item interactions, likelihood-based evaluation, and the assessment of properties such as calibration (Guo et al., 2017; Kweon et al., 2022). We therefore ask whether this probabilistic reformulation can retain the ranking effectiveness of the original recommender while enabling maximum likelihood training without sampled negative items in the objective.

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.

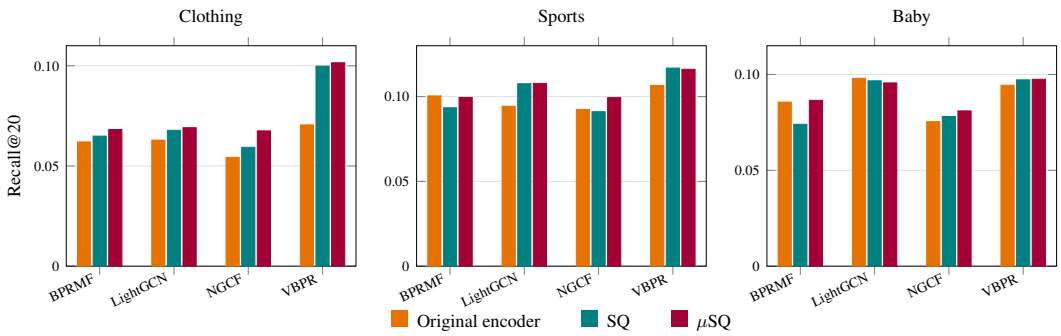

Figure 1: Recall@20 of four recommenders using the original models, the squared circuit (SQ), and the monotone-squared circuit (μSQ), on three of the five datasets; Table 1 reports all five. Both circuits are trained by joint maximum likelihood.

Building on this, we introduce *CiRe*, a framework for *circuitising* (i.e., converting to PCs) existing factorisation-based recommenders: we retain the encoder's representation-learning design, replace only its final factorised interaction with a tractable probabilistic circuit, and train the resulting model end-to-end. We consider squared and monotone-squared circuit constructions, which turn the original interaction into a nonnegative user–item mass while preserving tractable normalisation. For both families, the required normalisers reduce to compact Gram-matrix contractions of the learned representations. The circuitised recommender therefore defines a tractable joint distribution over user–item interactions and can be trained by maximum likelihood without negative sampling in the principal recommendation objective.

Across nine recommender architectures and five datasets, CiRe frequently matches or improves the top-k ranking performance of the original models (see Figure 1). Our experiments identify the circuitisation recipe as the most consistent contributor to these gains, while neither circuit parametrisation nor likelihood training alone consistently achieves the strongest results. Furthermore, recognising that recommendation is a discriminative task, we also train CiRe by conditional maximum likelihood; its ranking performance is broadly in line with joint training, with no systematic advantage for either formulation. We further evaluate more structured circuit variants, but find that additional circuit complexity improves held-out likelihood more consistently than ranking. Overall, a simple probabilistic reparametrisation of the final interaction, combined with maximum-likelihood training, equips existing recommenders with normalised probabilities and probabilistic queries while retaining or improving their ranking performance. For simple encoders, removing negative sampling from recommendation objectives can additionally reduce training cost.

Our main contributions are: **C1)** We introduce *CiRe*, a framework for converting the final factorised interaction of existing recommenders into a tractable probabilistic model while retaining their representation-learning design. **C2)** Building on established factorisation–circuit constructions, we adapt squared probabilistic circuits to recommendation and derive compact Gram-matrix forms for their joint and user-conditional normalisers, enabling maximum-likelihood training without negative sampling in the principal recommendation objective. **C3)** Across nine recommender architectures and five datasets, we show that CiRe frequently preserves or improves top-k ranking, identify circuitisation recipe as a consistent contributor to these gains, and characterise how joint versus conditional training, circuit structure, and encoder complexity affect accuracy, likelihood and computational cost.

2 Circuitising Factorisation-Based Recommenders

We introduce *CiRe*, a recipe for converting factorisation-based recommenders into tractable probabilistic models, illustrated in Figure 2. CiRe retains the encoder's representation-learning design and replaces its final factorised interaction with a nonnegative circuit whose normaliser can be evaluated exactly. The encoder and circuit parameters are trained jointly by maximum likelihood.

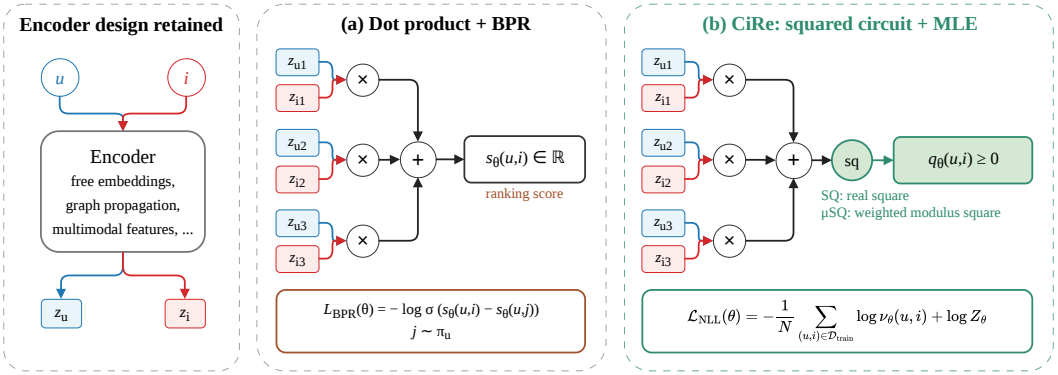

Figure 2: Overview of *CiRe*. The encoder is retained, while the factorised score is replaced by an SQ or μSQ circuit interaction with exact Gram-based normalisation. The model is trained by maximum likelihood without negative sampling in the principal objective.

Factorisation-based recommendation Let $\mathcal{U}$ and $\mathcal{I}$ denote sets of U users and I items, and let $\mathcal{D}_{\text{tr}} \subseteq \mathcal{U} \times \mathcal{I}$ contain the observed interactions. We consider recommenders whose encoder produces user and item representations $e_u, e_i \in \mathbb{R}^d$, combined through

$$s_\theta(u,i) = e_u^\top e_i = \sum_{r=1}^{d} e_{ur} e_{ir}. \tag{1}$$

In classical matrix factorisation these representations are learned directly as embeddings (Koren et al., 2009), whereas graph- and content-based recommenders derive them through interaction propagation or side information (He et al., 2020; He & McAuley, 2016). Despite these different representation-learning mechanisms, their final interaction can retain the same factorised form.

A common training objective is Bayesian Personalized Ranking (BPR) (Rendle et al., 2009). Given $(u,i) \sim \text{Unif}(\mathcal{D}_{\text{tr}})$ and an alternative item $j \sim \pi_u$, BPR minimises

$$\mathcal{L}_{\text{BPR}}(\theta) = -\mathbb{E}_{(u,i),j} \log \sigma(s_\theta(u,i) - s_\theta(u,j)), \tag{2}$$

where σ denotes the logistic function. BPR maximises the likelihood of pairwise preference relations constructed from sampled alternatives, instead of the likelihood of a distribution over user–item interactions. Regularisation and model-specific auxiliary losses may be added to this objective.

Circuitising the factorised interaction A *circuit* is a directed acyclic computational graph composed of input, sum, and product units (Vergari et al., 2021). Each unit has a *scope*, the set of variables on which its output depends. A circuit is *smooth* when the inputs to each sum unit have the same scope and *decomposable* when the inputs to each product unit have disjoint scopes. With tractably summable input functions, these properties allow sums over variables to be computed efficiently. A *probabilistic circuit* (PC) is a circuit whose output is a nonnegative function $q(\mathbf{x})$. If its total mass is finite and positive, it defines $p(\mathbf{x}) = q(\mathbf{x})/Z$, with $Z = \sum_{\mathbf{x}} q(\mathbf{x})$.

Different circuit families enforce nonnegativity in different ways. These constructions connect naturally to factorised models. Matrix and tensor factorisations admit equivalent sum–product circuit representations because their entries are computed through sums and products of lower-dimensional factors (Loconte et al., 2025a). In particular, the recommender interaction in Equation (1) is a shallow smooth and decomposable circuit: each term combines a user-specific and an item-specific factor, while the outer sum aggregates the latent dimensions. With unrestricted embeddings, however, its output may be negative and therefore cannot directly represent probability mass.

Building on this connection, CiRe replaces the final factorised interaction with one of two circuit families: a squared interaction (SQ) and a monotone sum-of-compatible-squares interaction (μSOCS). We instantiate both through complex-valued adapters applied to the encoder outputs, producing $z_u, z_i \in \mathbb{C}^d$. Complex parametrisations are convenient for these constructions as the modulus square of a complex circuit is real and nonnegative, with $|c(\mathbf{x})|^2 = (\text{Re}\, c(\mathbf{x}))^2 + (\text{Im}\, c(\mathbf{x}))^2$, linking complex squared circuits to sums of compatible real squares (Loconte et al., 2025b).

The SQ interaction squares the real part of the complex factorised interaction:

$$q_{\mathrm{SQ}}(u,i) = \left[\mathrm{Re}\sum\nolimits_{r=1}^{d} z_{ur}\overline{z_{ir}}\right]^2. \tag{3}$$

For μSOCS, we partition the d complex dimensions into S disjoint blocks $\{\mathcal{B}_s\}_{s=1}^{S}$, with S dividing d, and introduce user and item weights $g_u, h_i \in \mathbb{R}^K$:

$$q_{\mu}^{S,K}(u,i) = \left[\sum\nolimits_{k=1}^{K} e^{g_{uk}+h_{ik}}\right]\left[\sum\nolimits_{s=1}^{S}\left|\sum\nolimits_{r\in\mathcal{B}_s} z_{ur}z_{ir}\right|^2\right]. \tag{4}$$

Here, S controls the number of modulus-square components and K the rank of the monotonic factor. Our main configuration sets $S = K = 1$, yielding

$$q_{\mu\mathrm{SQ}}(u,i) = e^{g_u+h_i}\left|\sum\nolimits_{r=1}^{d} z_{ur}z_{ir}\right|^2, \tag{5}$$

which we denote μSQ. The main experiments compare SQ and μSQ, while additional experiments vary S and K to study the effect of circuit structure. Both interactions yield a nonnegative mass, which we normalise below to obtain the distributions used for learning and recommendation.

Learning the interaction distribution We use joint maximum likelihood as the main formulation of CiRe. Normalising q_θ over all user–item pairs defines

$$p_\theta(u,i) = \frac{q_\theta(u,i)}{Z_\theta}, \qquad Z_\theta = \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}} q_\theta(u,i). \tag{6}$$

For $N = |\mathcal{D}_{\mathrm{tr}}|$ observed interactions, the average negative log-likelihood is

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = -\frac{1}{N}\sum\nolimits_{(u,i)\in\mathcal{D}_{\mathrm{tr}}} \log q_\theta(u,i) + \log Z_\theta. \tag{7}$$

The first term increases the mass assigned to observed interactions, whereas the partition function accounts for mass over the complete user–item support. During minibatch training, observed interactions are sampled from $\mathcal{D}_{\mathrm{tr}}$, while Z_θ is evaluated exactly using the contractions derived below. The principal recommendation objective therefore does not require sampled negative items. At recommendation time the user is observed, so conditioning the joint model gives

$$p_\theta(i \mid u) = \frac{q_\theta(u,i)}{Z_\theta(u)}, \qquad Z_\theta(u) = \sum_{i\in\mathcal{I}} q_\theta(u,i). \tag{8}$$

For a fixed user, $Z_\theta(u)$ is constant across candidate items; ranking by $p_\theta(i \mid u)$ is therefore equivalent to ranking by $q_\theta(u,i)$. We also consider conditional maximum likelihood, which directly fits the item distribution given each user. If n_u denotes the number of training interactions involving user u,

$$\mathcal{L}_{\mathrm{CNLL}}(\theta) = -\frac{1}{N}\sum\nolimits_{(u,i)\in\mathcal{D}_{\mathrm{tr}}} \log q_\theta(u,i) + \sum\nolimits_{u\in\mathcal{U}} \frac{n_u}{N}\log Z_\theta(u). \tag{9}$$

The distinction from joint training is precisely the user marginal. Since $p_\theta(u) = Z_\theta(u)/Z_\theta$,

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = \mathcal{L}_{\mathrm{CNLL}}(\theta) - \sum\nolimits_{u\in\mathcal{U}} \frac{n_u}{N}\log p_\theta(u). \tag{10}$$

Joint maximum likelihood therefore additionally fits the empirical user-frequency distribution. Because these frequencies can reflect exposure and data-collection processes as well as preference (Schnabel et al., 2016), we assess empirically whether modelling this marginal affects recommendation performance. The complete training criterion is

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{NLL}}(\theta) + \lambda\Omega(\theta) + \mathcal{L}_{\mathrm{aux}}(\theta), \tag{11}$$

with $\mathcal{L}_{\mathrm{NLL}}$ replaced by $\mathcal{L}_{\mathrm{CNLL}}$ in the conditional experiments. Encoder-specific auxiliary objectives are retained when present and may themselves use contrasting examples.

Exact normalisation through Gram matrices The likelihoods above require summing the interaction mass over all items, or over all user–item pairs in the joint case. For the squared interactions used by CiRe, these sums can be arranged into compact Gram-matrix contractions, specialising standard circuit marginalisation operations to the user–item setting (Loconte et al., 2023; Vergari et al., 2021). For SQ, define $H_U = \sum_{u\in\mathcal{U}} \widetilde{z}_u \widetilde{z}_u^\top$, $H_I = \sum_{i\in\mathcal{I}} \widetilde{z}_i \widetilde{z}_i^\top$. Using $q_{\mathrm{SQ}}(u,i) = (\widetilde{z}_u^\top \widetilde{z}_i)^2$, the joint and conditional normalisers are

$$Z_{\mathrm{SQ}} = \sum_{r,t=1}^{2d} (H_U)_{rt}(H_I)_{rt}, \qquad Z_{\mathrm{SQ}}(u) = \widetilde{z}_u^\top H_I \widetilde{z}_u. \tag{12}$$

The joint normaliser therefore contracts summaries over all users and items, while the conditional normaliser reuses the same item Gram across users. For μSQ, the monotonic factors are absorbed into weighted complex Gram matrices, $G_U = \sum_{u\in\mathcal{U}} e^{g_u} z_u \overline{z_u}^\top$, $G_I = \sum_{i\in\mathcal{I}} e^{h_i} z_i \overline{z_i}^\top$. The corresponding normalisers are

$$Z_\mu = \sum_{r,t=1}^{d} (G_U)_{rt}(G_I)_{rt}, \qquad Z_\mu(u) = e^{g_u} z_u^\top G_I \overline{z_u}. \tag{13}$$

These expressions are exactly the sums of the nonnegative μSQ masses over the catalogue.

3 Experimental Setup

This section presents the experimental setup and the results of the circuitised recommender systems. In particular, we aim to answer to the following Research Questions (RQs):

RQ1: *Can factorisation-based recommenders be circuitised into tractable probabilistic models while retaining or improving recommendation effectiveness?*

RQ2: *How do joint and conditional maximum-likelihood training differ in ranking performance and probabilistic modelling?*

RQ3: *What drives the ranking changes of CiRe: maximum-likelihood training, the circuit interaction, or their combination?*

RQ4: *How does circuit structure affect ranking, held-out likelihood, and computational cost?*

Datasets We evaluate on five datasets spanning different recommendation domains. Following prior multimodal recommendation studies (Zhang et al., 2021; Yu et al., 2025), we consider three categories from Amazon Reviews (McAuley et al., 2015): *Baby*, *Sports and Outdoors (Sports)*, and *Clothing, Shoes, and Jewelry (Clothing)*. To assess generalisation beyond the Amazon domains, we additionally include DBbook and MovieLens-1M (ML-1M) (Spillo et al., 2025). Dataset statistics and relevance thresholds are reported in Appendix A.

Models We consider nine recommender architectures spanning classical factorisation, graph-based collaborative filtering, self-supervised learning, and multimodal recommendation. BPRMF (Rendle et al., 2009) serves as the latent-factor baseline; NGCF (Wang et al., 2019) and LightGCN (He et al., 2020) represent graph-based collaborative filtering; and SGL (Wu et al., 2021) augments graph recommendation with self-supervised learning. For multimodal recommendation, we include VBPR (He & McAuley, 2016), BM3 (Zhou et al., 2023), FREEDOM (Zhou & Shen, 2023), PGL (Yu et al., 2025), and SLMRec (Tao et al., 2023). Each encoder is evaluated in its original form and with the SQ and μSQ circuit interactions introduced in Section 2.

Metrics and implementation We report Recall and nDCG at a cutoff of 20, ranking over the full eligible catalogue rather than a sampled candidate set; items observed during training are excluded at evaluation time.For each model, we evaluate the same 30-point hyperparameter grid over learning rate, embedding dimension, and regularisation strength. Models are trained for up to 200 epochs with batch size 1024, and both the hyperparameter configuration and checkpoint are selected by validation Recall@20. Visual and textual features are obtained with ResNet-50 (He et al., 2016) and MPNet (Song et al., 2020), respectively, following established multimodal recommendation practice (Attimonelli et al., 2026) and using Ducho (Attimonelli et al., 2024). Original and circuitised recommenders are implemented within Elliot (Anelli et al., 2021) using PyTorch. Experiments

Table 1: Ranking performance in Recall@20 (R@20) and nDCG@20 (N@20) for the original recommenders (–), SQ, and μSQ across nine encoders and five datasets. Best results within each encoder–dataset pair are shown in **bold** and second-best results are underlined; * denotes statistical significant improvements of best circuit over base recommender ($p < 0.05$).

		Clothing		DBbook		Sports		Baby		ML-1M	
Encoder	Model	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
BPRMF	–	0.0620	0.0281	0.2637	0.1476	**0.1004**	0.0462	0.0856	0.0383	0.2997	0.2393
	SQ	0.0649	0.0304	0.2579	0.1463	0.0934	0.0448	0.0740	0.0342	0.3019	**0.2403**
	μSQ	**0.0682***	**0.0320***	**0.2668***	**0.1522***	0.0995	**0.0478**	**0.0865***	**0.0395***	**0.3044**	0.2368
LightGCN	–	0.0629	0.0282	0.2622	0.1403	0.0943	0.0437	**0.0980**	0.0428	0.2890	0.2304
	SQ	0.0678	0.0314	**0.2695***	**0.1460***	0.1076	0.0495	0.0967	**0.0436***	0.2980	**0.2360***
	μSQ	**0.0691***	**0.0323***	0.2625	0.1408	**0.1078***	**0.0514***	0.0956	0.0427	**0.3025***	0.2357
NGCF	–	0.0543	0.0246	0.2562	0.1396	0.0924	0.0423	0.0754	0.0332	0.3013	0.2378
	SQ	0.0593	0.0292	0.2591	0.1458	0.0911	0.0420	0.0781	0.0335	0.3027	0.2379
	μSQ	**0.0675***	**0.0317***	**0.2672***	**0.1499***	**0.0994***	**0.0478***	**0.0810***	**0.0384***	**0.3047**	**0.2385**
VBPR	–	0.0705	0.0320	0.2686	0.1507	0.1066	0.0493	0.0944	0.0425	**0.3014**	**0.2402**
	SQ	0.0998	0.0466	**0.2696***	**0.1601***	**0.1168***	0.0542	0.0972	**0.0434***	0.2881	0.2255
	μSQ	**0.1015***	**0.0471***	0.2684	0.1590	0.1161	**0.0553***	**0.0975***	0.0433	0.2942	0.2325
BM3	–	**0.0711***	**0.0326***	**0.2586**	**0.1407***	0.0985	0.0453	**0.0960**	**0.0432**	0.1748	0.1467
	SQ	0.0675	0.0316	0.2292	0.1296	0.0961	0.0463	0.0803	0.0352	0.2912	0.2316
	μSQ	0.0678	0.0318	0.2459	0.1312	**0.0991**	**0.0480***	0.0823	0.0352	**0.2956***	**0.2337***
FREEDOM	–	0.1019	0.0463	**0.2926**	**0.1649**	0.1161	0.0519	0.1049	0.0463	0.2717	0.2154
	SQ	**0.1073***	**0.0493***	0.2825	0.1586	0.1232	0.0567	0.1075	0.0474	0.2981	0.2409
	μSQ	0.1049	0.0481	0.2881	0.1567	**0.1250***	**0.0571***	**0.1105***	**0.0482***	**0.2992***	**0.2439***
PGL	–	0.0913	0.0410	**0.2795**	0.1476	0.1142	0.0517	0.1046	0.0461	0.2934	0.2436
	SQ	**0.1050***	**0.0487***	0.2646	**0.1622***	**0.1238***	**0.0569***	0.1016	0.0462	**0.3045***	**0.2479***
	μSQ	0.0997	0.0456	0.2723	0.1540	0.1206	0.0551	**0.1100***	**0.0481***	0.3043	0.2449
SLMRec	–	0.0700	0.0320	0.2233	0.1318	0.0965	0.0458	0.0742	0.0345	0.1844	0.1285
	SQ	0.0729	**0.0341***	**0.2365***	**0.1333***	0.0955	0.0423	0.0914	**0.0387***	**0.2924***	**0.2320***
	μSQ	**0.0732***	0.0336	0.2273	0.1322	**0.1017***	**0.0463***	**0.0916***	**0.0387***	0.2896	0.2284
SGL	–	0.0652	0.0303	0.2652	0.1546	0.1066	0.0509	0.0770	0.0373	0.2810	0.2201
	SQ	0.0675	0.0315	**0.2730***	**0.1589***	**0.1074**	**0.0519**	0.0849	**0.0411***	**0.2970***	**0.2442***
	μSQ	**0.0701***	**0.0327***	0.2714	0.1565	0.1051	0.0489	**0.0852***	0.0403	0.2954	0.2429

are run on a single compute node equipped with a NVIDIA A100 GPU. Complete hyperparameter configurations and additional implementation details are provided in Appendix A.

4 Results

4.1 Circuitisation preserves or improves recommendation effectiveness

Table 1 compares each original recommender with its SQ and μSQ counterparts trained by joint maximum likelihood. A circuitised variant attains the highest Recall@20 in 37 of 45 encoder–dataset pairs: μSQ is best in 25 cases and SQ in 12, while the original recommender is best in 8. For each encoder–dataset pair, statistical significance is assessed between the best-performing circuitised variant and the corresponding original recommender using a paired t-test over per-user metric values derived from the recommendation lists (Malitesta et al., 2025). Significant improvements are marked with * ($p < 0.05$). Thus, circuitisation generally preserves and often improves the ranking effectiveness of the underlying recommender. The gains are particularly consistent for the simpler factorised and graph-based models. Across BPRMF, LightGCN, NGCF, and VBPR, a circuitised variant achieves the highest Recall@20 in 17 of 20 settings. The effect can be substantial: on Clothing, for example, VBPR improves from 0.0705 to 0.1015, while BPRMF, LightGCN, and NGCF also improve under circuitisation. More elaborate architectures show a less uniform pattern: BM3 is the clearest counterexample, whereas FREEDOM, PGL, SLMRec, and SGL benefit on several datasets. The effect of circuitisation therefore depends on the representations supplied by the underlying encoder rather than on one circuit head being uniformly preferable.

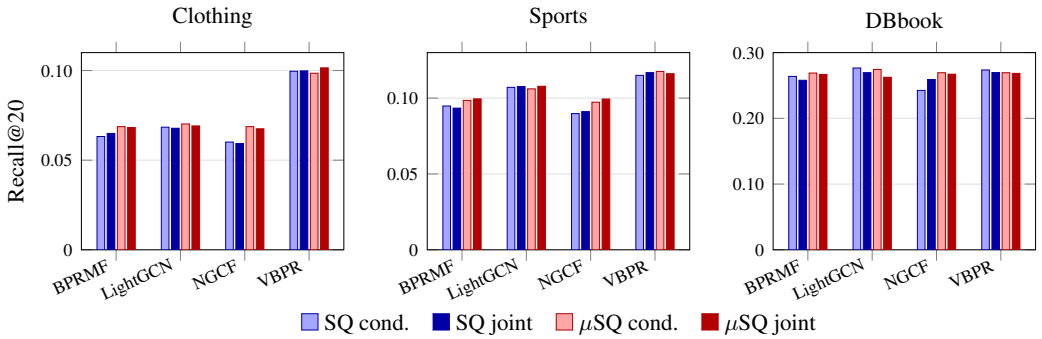

Figure 3: Recall@20 under conditional and joint maximum likelihood for SQ and μSQ on representative encoder–dataset pairs. Neither formulation exhibits a systematic advantage.

Circuitisation also has architecture-dependent computational impact. Across complete epochs, μSQ is faster on the median dataset for BPRMF, LightGCN, NGCF, and VBPR, but slower for the remaining encoders: for simple models, the removal of negative sampling can outweigh the cost of exact normalisation. For BPRMF, for instance, negative sampling alone costs 8.19 ms per batch, whereas the complete circuitised step costs 2.3 ms. This advantage disappears when graph propagation or auxiliary objectives dominate computation. Complete timing results are reported in Appendix D.

We additionally evaluate probability calibration, pairing observed interactions with sampled unobserved items and comparing predicted scores with empirical binary frequencies. Under min–max normalisation, at least one circuitised variant obtains lower calibration error than its matched original recommender in seven of nine encoder–dataset settings. Full results are reported in Appendix E.

*These experiments answer **RQ1**: factorisation-based recommenders can be circuitised into tractable probabilistic models while retaining, and frequently improving, top-k recommendation effectiveness. Gains are most consistent for simple factorised and graph-based encoders, for which circuitisation can also reduce training cost, while supporting probabilistic queries and competitive calibration.*

4.2 Joint and conditional likelihood yield similar ranking performance

Because recommendation ranks items conditional on a user, we compare joint maximum likelihood in Equation (7) with conditional maximum likelihood in Equation (9), which directly models $p_\theta(i \mid u)$. Figure 3 illustrates the comparison on representative settings, while complete conditional results are reported in Appendix B. Circuitisation remains competitive under conditional training: a circuitised variant achieves the highest Recall@20 in 34 of 45 encoder–dataset pairs, including 15 of 20 for BPRMF, LightGCN, NGCF, and VBPR. The gains observed in Section 4.1 are therefore not specific to joint modelling.

Across the 90 encoder–dataset–head comparisons, joint training yields higher Recall@20 in 48 cases and conditional training in 42, with no significant overall difference (Wilcoxon signed-rank, $p = 0.50$). The mean joint-minus-conditional difference is $+0.0007$, and 51 of the 90 absolute differences are below 0.0025. The split is similarly balanced within each circuit family, with joint training winning 22 of 45 SQ comparisons and 26 of 45 μSQ comparisons. We therefore find no systematic ranking advantage for either objective.

Their distinction is probabilistic. Since $\log p_\theta(u, i) = \log p_\theta(i \mid u) + \log p_\theta(u)$, joint maximum likelihood additionally models the user marginal $p_\theta(u)$, whereas conditional maximum likelihood fits only $p_\theta(i \mid u)$. For a fixed user, this marginal is constant across candidate items and hence does not directly affect their ordering, although it can influence the shared representations during training. Empirically, this additional modelling does not translate into consistent top-k gains.

*Addressing **RQ2**, joint and conditional maximum likelihood perform similarly for ranking, while differing in probabilistic scope: conditional training models the item distribution required for recommendation, while joint training additionally provides the user marginal and user–item distribution.*

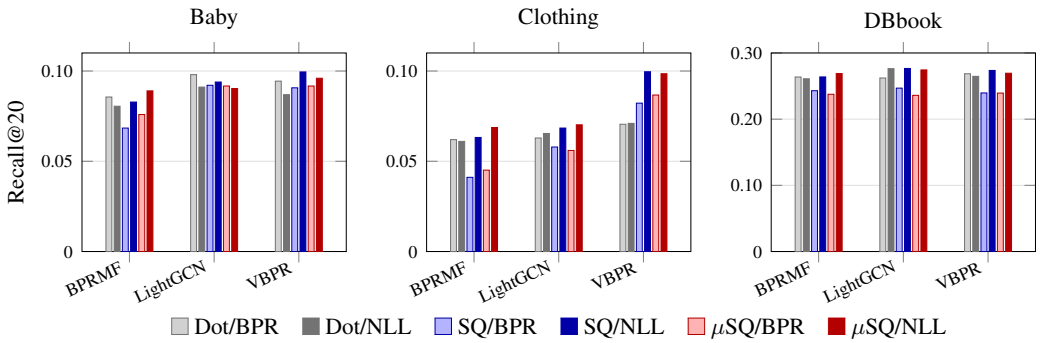

Figure 4: Recall@20 across interaction functions and training objectives.

4.3 Ranking gains concentrate in the circuitisation recipe

To identify which components of CiRe are associated with the ranking gains, we compare three interactions—Dot, SQ, and μSQ—under BPR and conditional maximum likelihood for BPRMF, LightGCN, and VBPR on Baby, Clothing, and DBbook.

Dot+BPR corresponds to the original recommender configuration. For Dot+NLL, the conventional dot-product scores are interpreted as logits of a categorical distribution over the complete item catalogue. The normalisation is evaluated over all items, so Dot+NLL performs exact conditional maximum-likelihood training. SQ+NLL and μSQ+NLL optimise the same conditional likelihood, but compute the catalogue normaliser through the tractable circuit contractions derived in Section 2. Each interaction–objective configuration is retrained independently using the same hyperparameter search space and selection protocol. Figure 4 summarises the resulting trends, with complete results reported in Appendix C.1.

The effect of the training objective strongly depends on the interaction. Replacing BPR with conditional maximum likelihood improves Recall@20 in only 2 of 9 Dot comparisons, but in all 9 SQ comparisons and 8 of 9 μSQ comparisons. Conditional likelihood is therefore not consistently beneficial for the conventional dot product: its gains are concentrated in the circuit interactions.

The converse comparison shows the same pattern. Under BPR, replacing Dot with SQ or μSQ improves only 2 of 18 comparisons. Under conditional likelihood, however, a circuit interaction outperforms Dot+NLL in 17 of 18 comparisons. Consequently, SQ+NLL or μSQ+NLL achieves the highest Recall@20 in 8 of 9 encoder–dataset settings, with Dot+BPR best only for LightGCN on Baby. The strongest ranking results are therefore associated with the complete circuitisation recipe: combining the circuit interaction with likelihood training.

*Addressing **RQ3**, neither conditional likelihood nor circuit parametrisation is consistently beneficial. Their combination achieves the best result in eight of nine controlled settings, supporting the effectiveness of the circuitisation recipe.*

4.4 Additional squared structure benefits likelihood more than ranking

We examine how the structure of the circuit head affects recommendation and distribution modelling by varying the number of squared components S and the monotonic-factor parameterisation K in Equation (4). Complete results are reported in Appendix C.2.

For $S \in \{1, 2, 4\}$, we keep the total embedding width fixed at $d = 256$, so increasing S repartitions the representation rather than increasing its width. Ranking consistently favours the simpler structure: $S = 1$ attains the highest Recall@20 in eight of the nine encoder–dataset settings and outperforms the larger-S alternatives in 16 of 18 pairwise comparisons. The only exception is LightGCN on Baby, where a larger-S configuration performs best. Increasing S therefore provides no systematic benefit for top-k recommendation.

Held-out likelihood shows a different pattern. Relative to $S = 1$, additional squared components reduce test NLL in four of six LightGCN comparisons and three of six VBPR comparisons, but in none of the six BPRMF comparisons. The broader (S, K) grid reinforces this distinction: the

best-ranking configuration uses $S = 1$ in eight of nine encoder–dataset settings, whereas the best-NLL configuration uses $S > 1$ in all six LightGCN and VBPR settings. Thus, additional squared structure can improve the fit of the conditional distribution without improving the ordering of its highest-ranked items.

Varying the monotonic factor yields weaker effects. With $S = 1$, $K = 1$ achieves the highest Recall in seven of nine settings, and increasing K provides no consistent ranking improvement.

*Addressing **RQ4**, additional circuit structure does not systematically improve ranking. Additional squared structure improves held-out likelihood more consistently, indicating that the structure useful for modelling the full distribution need not coincide with that required for top-k recommendation.*

5 Related work

Recommendation objectives and representation learning Much of recommender-system research focuses on improving user and item representations through graph propagation, self-supervision, or multimodal information (Wang et al., 2019; He et al., 2020; Wu et al., 2021; He & McAuley, 2016; Zhou et al., 2023; Zhou & Shen, 2023; Yu et al., 2025; Tao et al., 2023). Many such models nevertheless retain a simple factorised interaction and are trained with BPR, which maximises the likelihood of sampled pairwise preference events (Rendle et al., 2009). Alternatives to sampled pairwise training include confidence-weighted implicit-feedback factorisation, non-sampling factorisation machines, and multinomial generative recommenders (Hu et al., 2008; Chen et al., 2020; Liang et al., 2018). CiRe is complementary to these directions: it retains the encoder's representation-learning design and instead turns the final factorised interaction into a tractably normalised probabilistic model, enabling joint or conditional maximum-likelihood training.

From factorisations to probabilistic circuits Probabilistic circuits build on a broader line of work on circuits and tractable probabilistic models, where structured sum–product representations enable exact probabilistic inference (Darwiche, 2003; Vergari et al., 2019; Choi et al., 2020; Vergari et al., 2021). In parallel, low-rank tensor factorisations provide a classical framework for representing high-dimensional interactions through compact multilinear structure (Kolda & Bader, 2009). Recent work connects these two views by showing that tensor factorisations admit equivalent circuit representations and that signed factorised predictors can be converted into nonnegative, tractably normalisable models through squared constructions (Loconte et al., 2025a; 2023). Subsequent work studies more general squared and sum-of-squares circuit families and their representational properties (Loconte et al., 2024; 2025b). We build on these constructions: our contribution is to transfer the factorisation–circuit connection to recommendation, use the circuit as a replaceable interaction head over heterogeneous encoders, and study how maximum-likelihood training and circuit structure affect top-k recommendation and probabilistic modelling.

6 Conclusion

We introduced *CiRe*, a recipe for converting factorisation-based recommenders into tractable probabilistic models. CiRe retains the encoder's representation-learning design, replaces the final factorised interaction with a nonnegative circuit head, and enables maximum-likelihood training with exact normalisation. Across nine architectures and five datasets, circuitisation frequently preserves or improves top-k recommendation effectiveness and, for simpler encoders, can also reduce training cost relative to sampled BPR. Our experiments show that neither likelihood training nor circuits reparametrisation alone is consistently beneficial: instead, the strongest results are obtained when circuit interactions are combined with likelihood training. Joint and conditional maximum likelihood yield similar ranking performance despite differing in probabilistic scope. Additional circuit structure does not systematically improve ranking, although additional squared structure improves held-out likelihood more consistently. Future work can extend this analysis to other tractable circuit structures and alternative recommendation paradigms, such as sequential recommendation. Overall, CiRe shows that the shared factorised interaction of existing recommenders can support tractable distribution learning, exact probabilistic queries, and competitive recommendation performance without redesigning their representation-learning machinery.

AI Disclosure Statement

Generative AI assistants were used only for language polishing. The authors are responsible for all claims, results, and the final manuscript.

References

Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, and Tommaso Di Noia. Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation. In *SIGIR*, pp. 2405–2414, 2021. doi: 10.1145/3404835.3463245.

Matteo Attimonelli, Danilo Danese, Daniele Malitesta, Claudio Pomo, Giuseppe Gassi, and Tommaso Di Noia. Ducho 2.0: Towards a more up-to-date unified framework for the extraction of multimodal features in recommendation. In *WWW Companion*, pp. 1075–1078, 2024.

Matteo Attimonelli, Danilo Danese, Angela Di Fazio, Daniele Malitesta, Claudio Pomo, and Tommaso Di Noia. Large-scale benchmarks for multimodal recommendation with ducho. *Expert Syst. Appl.*, 307:130813, 2026.

Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Efficient non-sampling factorization machines for optimal context-aware recommendation. In *WWW*, pp. 2400–2410, 2020. doi: 10.1145/3366423.3380303.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic modeling. Technical report, University of California, Los Angeles (UCLA), 2020.

Adnan Darwiche. A differential approach to inference in bayesian networks. *Journal of the ACM (JACM)*, 50:280–305, 2003.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *ICML*, pp. 1321–1330, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778. IEEE Computer Society, 2016.

Ruining He and Julian J. McAuley. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI*, pp. 144–150, 2016. doi: 10.1609/aaai.v30i1.9973.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *SIGIR*, pp. 639–648, 2020. doi: 10.1145/3397271.3401063.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pp. 263–272, 2008. doi: 10.1109/ICDM.2008.22.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems - An Introduction*. Cambridge University Press, 2010.

Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51 (3):455–500, 2009. doi: 10.1137/07070111X.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *KDD*, pp. 1748–1757, 2020. doi: 10.1145/3394486.3403226.

Wonbin Kweon, SeongKu Kang, and Hwanjo Yu. Obtaining calibrated probabilities with personalized ranking models. In *AAAI*, pp. 4083–4091. AAAI Press, 2022.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pp. 689–698, 2018. doi: 10.1145/3178876.3186150.

Lorenzo Loconte, Nicola Di Mauro, Robert Peharz, and Antonio Vergari. How to turn your knowledge graph embeddings into generative models. In *NeurIPS*, 2023.

Lorenzo Loconte, Aleksanteri M. Sladek, Stefan Mengel, Martin Trapp, Arno Solin, Nicolas Gillis, and Antonio Vergari. Subtractive mixture models via squaring: Representation and learning. In *ICLR*, 2024.

Lorenzo Loconte, Antonio Mari, Gennaro Gala, Robert Peharz, Cassio de Campos, Erik Quaeghebeur, Gennaro Vessio, and Antonio Vergari. What is the relationship between tensor factorizations and circuits (and how can we exploit it)? *Trans. Mach. Learn. Res.*, 2025, 2025a.

Lorenzo Loconte, Stefan Mengel, and Antonio Vergari. Sum of squares circuits. In *AAAI*, pp. 19077–19085, 2025b. doi: 10.1609/AAAI.V39I18.34100.

Daniele Malitesta, Giandomenico Cornacchia, Claudio Pomo, Felice Antonio Merra, Tommaso Di Noia, and Eugenio Di Sciascio. Formalizing multimedia recommendation through multimodal deep learning. *Trans. Recomm. Syst.*, 3(3):37:1–37:33, 2025.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pp. 43–52. ACM, 2015.

Steffen Rendle and Christoph Freudenthaler. Improving pairwise learning for item recommendation from implicit feedback. In *WSDM*, pp. 273–282, 2014. doi: 10.1145/2556195.2556248.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pp. 452–461, 2009.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*, volume 48, pp. 1670–1679, 2016.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS*, 2020.

Giuseppe Spillo, Elio Musacchio, Cataldo Musto, Marco de Gemmis, Pasquale Lops, and Giovanni Semeraro. See the movie, hear the song, read the book: Extending movielens-1m, last.fm-2k, and dbbook with multimodal data. In *RecSys*, pp. 847–856. ACM, 2025.

Zhulin Tao, Xiaohao Liu, Yewei Xia, Xiang Wang, Lifang Yang, Xianglin Huang, and Tat-Seng Chua. Self-supervised learning for multimedia recommendation. *IEEE Trans. Multimedia*, 25: 5107–5116, 2023.

Antonio Vergari, Nicola Di Mauro, and Guy Van den Broeck. Tractable probabilistic models: Representations, algorithms, learning, and applications. *Tutorial at the 35th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *NeurIPS*, pp. 13189–13201, 2021.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, pp. 165–174, 2019. doi: 10.1145/3331184.3331267.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, pp. 726–735, 2021. doi: 10.1145/3404835.3462862.

Penghang Yu, Zhiyi Tan, Guanming Lu, and Bing-Kun Bao. Mind individual information! principal graph learning for multimedia recommendation. In *AAAI*, pp. 13096–13105, 2025.

Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Shu Wu, Shuhui Wang, and Liang Wang. Mining latent structures for multimedia recommendation. In *ACM Multimedia*, pp. 3872–3880. ACM, 2021.

Xin Zhou and Zhiqi Shen. A tale of two graphs: Freezing and denoising graph structures for multimodal recommendation. In *ACM MM*, pp. 935–943, 2023. doi: 10.1145/3581783.3611943.

Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. Bootstrap latent representations for multi-modal recommendation. In *WWW*, pp. 845–854, 2023. doi: 10.1145/3543507.3583251.

Appendix A Datasets and evaluation **14**

Appendix B Conditional-likelihood ranking results **15**

Appendix C Additional ablations **16**

C.1 Score–objective ablation 16

C.2 Circuit structure: monotonic rank and number of squares 16

Appendix D Time measurements **19**

Appendix E Calibration **21**

A DATASETS AND EVALUATION

Table 2 summarises the five datasets used in our experiments. The reported user and item counts are computed over the union of identities appearing in the train, validation, and test split files. Ratings at or above the relevance threshold τ are treated as relevant for ranking evaluation; all recorded training interactions are retained during training.

Table 2: Dataset statistics. User and item counts cover the union of identities across the train, validation, and test splits. The relevance threshold determines positive labels for ranking evaluation.

Dataset	Users	Items	Train	Valid	Test	Threshold
Clothing	39,385	22,498	191,934	39,067	39,990	4
DBbook	6,142	5,217	44,357	9,776	9,145	1
Sports	35,598	18,285	217,687	37,783	39,882	4
Baby	19,445	7,037	118,356	20,523	21,643	4
ML-1M	6,040	3,075	637,172	136,411	136,682	4

Ranking is performed over the full eligible catalogue rather than over sampled candidate sets, avoiding the evaluation distortions that candidate sampling can introduce (Krichene & Rendle, 2020). We use $\tau = 4$ for Baby, Clothing, Sports, and ML-1M, and $\tau = 1$ for DBbook. Multimodal models use the retained ResNet-50 (He et al., 2016) visual features and MPNet (Song et al., 2020) textual features.

Ranking metrics For user u, let $T_u = \{i : (u, i, r) \in \mathcal{D}_{\text{te}},\ r \geq \tau\}$, $\mathcal{U}_+ = \{u : |T_u| > 0\}$, and let $L_u = (i_{u1}, \ldots, i_{u20})$ denote the top-20 ranked items from the eligible catalogue. We report

$$\text{Recall@20} = \frac{1}{|\mathcal{U}_+|} \sum_{u \in \mathcal{U}_+} \frac{|L_u \cap T_u|}{|T_u|}, \tag{14}$$

$$\text{nDCG@20} = \frac{1}{|\mathcal{U}_+|} \sum_{u \in \mathcal{U}_+} \frac{\sum_{j=1}^{20} G_u(i_{uj}) / \log_2(j+1)}{\text{IDCG}_u@20}. \tag{15}$$

Held-out likelihood. We additionally evaluate the conditional likelihood of positive interactions. Let $C_u = [I] \setminus \mathcal{I}_u^{\text{tr}}$ be the recommendation-eligible catalogue for user u. We report

$$-\frac{1}{N} \sum_{(u,i) \in \mathcal{D}_{\text{te},+}} \log \frac{\nu(u,i)}{\sum_{j \in C_u} \nu(u,j)},$$

where N is the number of finite eligible positive interactions evaluated. The denominator is enumerated exactly over C_u in double precision. This metric averages over held-out interactions, whereas Recall@20 averages over users. It also differs from the training likelihood: training normalises over the full catalogue, while held-out evaluation excludes items already observed in training.

Search and selection Each recommender is trained by exploring learning rate in $\{10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 5 \cdot 10^{-3}, 10^{-2}\}$, embedding dimension in $\{64, 256\}$, and regularization in $\{0, 10^{-5}, 10^{-2}\}$. The product is 30 configurations. The search budget is 200 epochs, with batch size 1024 and seed 42. Structural encoder settings are held fixed within a comparison. Validation Recall@20 selects the configuration and epoch, and the selected epoch is replayed for test evaluation.

Table 3: **Conditional training.** Ranking performance in Recall@20 (R@20) and nDCG@20 (N@20) for the original recommenders (–), conditional SQ, and conditional μSQ across nine encoders and five datasets. Best results within each encoder–dataset pair are shown in **bold** and second-best results are underlined.

		Clothing		DBbook		Sports		Baby		ML-1M	
Encoder	Model	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
BPRMF	–	0.0620	0.0281	0.2637	0.1476	**0.1004**	0.0462	0.0856	0.0383	**0.2997**	**0.2393**
	SQ	0.0632	0.0296	0.2638	0.1513	0.0948	0.0445	0.0828	0.0381	0.2977	0.2337
	μSQ	**0.0687**	**0.0324**	**0.2689**	**0.1531**	0.0985	**0.0479**	**0.0890**	**0.0401**	0.2976	0.2339
LightGCN	–	0.0629	0.0282	0.2622	0.1403	0.0943	0.0437	**0.0980**	**0.0428**	0.2890	0.2304
	SQ	0.0684	0.0320	**0.2766**	**0.1533**	**0.1071**	0.0490	0.0939	0.0421	0.2962	0.2347
	μSQ	**0.0702**	**0.0328**	0.2744	0.1510	0.1061	**0.0509**	0.0903	0.0421	**0.3013**	**0.2380**
NGCF	–	0.0543	0.0246	0.2562	0.1396	0.0924	0.0423	0.0754	0.0332	**0.3013**	**0.2378**
	SQ	0.0601	0.0287	0.2425	0.1319	0.0898	0.0432	0.0747	0.0321	0.2967	0.2335
	μSQ	**0.0687**	**0.0322**	**0.2694**	**0.1453**	**0.0973**	**0.0475**	**0.0847**	**0.0393**	0.2998	0.2349
VBPR	–	0.0705	0.0320	0.2686	0.1507	0.1066	0.0493	0.0944	0.0425	**0.3014**	**0.2402**
	SQ	**0.0996**	**0.0466**	**0.2736**	**0.1560**	0.1150	0.0543	**0.0995**	**0.0438**	0.2949	0.2317
	μSQ	0.0985	0.0457	0.2694	0.1555	**0.1176**	**0.0556**	0.0960	0.0428	0.2909	0.2311
BM3	–	**0.0711**	**0.0326**	**0.2586**	**0.1407**	0.0985	0.0453	**0.0960**	**0.0432**	0.1748	0.1467
	SQ	0.0666	0.0314	0.2361	0.1316	0.0970	0.0458	0.0785	0.0346	0.2897	**0.2335**
	μSQ	0.0677	0.0318	0.2323	0.1267	**0.1002**	**0.0482**	0.0851	0.0361	**0.2910**	0.2334
FREEDOM	–	0.1019	0.0463	**0.2926**	**0.1649**	0.1161	0.0519	0.1049	0.0463	0.2717	0.2154
	SQ	**0.1047**	**0.0478**	0.2826	0.1614	0.1211	0.0555	0.1078	0.0478	0.3021	0.2440
	μSQ	0.1038	0.0473	0.2864	0.1592	**0.1247**	**0.0567**	**0.1103**	**0.0486**	**0.3044**	**0.2442**
PGL	–	0.0913	0.0410	**0.2795**	0.1476	0.1142	0.0517	**0.1046**	0.0461	0.2934	0.2436
	SQ	**0.1036**	**0.0481**	0.2742	**0.1634**	**0.1252**	**0.0568**	0.1009	0.0453	0.3083	**0.2503**
	μSQ	0.1007	0.0462	0.2778	0.1530	0.1214	0.0555	0.1044	**0.0468**	**0.3097**	0.2501
SLMRec	–	0.0700	0.0320	0.2233	**0.1318**	0.0965	0.0458	0.0742	0.0345	0.1844	0.1285
	SQ	**0.0722**	**0.0337**	0.2246	0.1174	**0.0992**	**0.0472**	**0.0931**	**0.0401**	**0.2965**	**0.2368**
	μSQ	0.0708	0.0331	**0.2255**	0.1278	0.0967	0.0458	0.0546	0.0230	0.2910	0.2338
SGL	–	0.0652	0.0303	0.2652	0.1546	0.1066	0.0509	0.0770	0.0373	0.2810	0.2201
	SQ	0.0678	0.0315	**0.2697**	**0.1590**	**0.1085**	**0.0524**	**0.0853**	0.0391	**0.2884**	0.2252
	μSQ	**0.0698**	**0.0326**	0.2685	0.1565	0.1074	0.0505	0.0848	**0.0400**	0.2874	**0.2266**

B CONDITIONAL-LIKELIHOOD RANKING RESULTS

This appendix reports the full conditional counterpart of Table 1: the same nine encoders and five datasets, with SQ and μSQ trained by conditional rather than joint maximum likelihood. Section 4.2 summarises the comparison; the two objectives agree on which encoder families benefit from circuitisation.

Table 4: Recall@20 for the controlled score–objective ablation. Best and second-best results within each encoder–dataset pair are shown in **bold** and underlined, respectively.

		Baby		Clothing		DBbook	
Encoder	Head	BPR	NLL	BPR	NLL	BPR	NLL
BPRMF	Dot	0.0856	0.0805	0.0620	0.0610	0.2637	0.2610
	SQ	0.0684	0.0828	0.0411	0.0632	0.2430	0.2638
	μSQ	0.0759	**0.0890**	0.0451	**0.0687**	0.2376	**0.2689**
LightGCN	Dot	**0.0980**	0.0911	0.0629	0.0654	0.2622	0.2763
	SQ	0.0921	0.0939	0.0579	0.0684	0.2469	**0.2766**
	μSQ	0.0917	0.0903	0.0560	**0.0702**	0.2360	0.2744
VBPR	Dot	0.0944	0.0869	0.0705	0.0710	0.2686	0.2647
	SQ	0.0907	**0.0995**	0.0822	**0.0996**	0.2396	**0.2736**
	μSQ	0.0917	0.0960	0.0867	0.0985	0.2394	0.2694

Table 5: Recall@20 varying the rank K of the monotonic factor of Equation (4). Best and second-best results within each encoder–dataset pair are shown in **bold** and underlined, respectively.

Encoder	Rank	Baby	Clothing	DBbook
BPRMF	$K=1$	**0.0890**	0.0687	**0.2689**
	$K=2$	0.0886	**0.0688**	0.2682
	$K=4$	0.0887	0.0687	0.2682
	$K=8$	0.0885	0.0673	0.2685
LightGCN	$K=1$	**0.0903**	**0.0702**	0.2744
	$K=2$	0.0858	0.0679	0.2731
	$K=4$	0.0831	0.0682	0.2663
	$K=8$	0.0845	0.0658	0.2604
VBPR	$K=1$	0.0960	0.0985	0.2694
	$K=2$	**0.0979**	0.0973	0.2689
	$K=4$	0.0969	0.0974	0.2684
	$K=8$	0.0973	0.0958	0.2692

C Additional ablations

C.1 Score–objective ablation

This subsection reports the full numbers behind Figure 4: the controlled score–objective ablation of Section 4.3, crossing the three interactions with the two training objectives on three encoders and three datasets.

C.2 Circuit structure: monotonic rank and number of squares

Monotonic rank Table 5 varies the rank K of the monotonic factor while keeping $S = 1$. Increasing K provides no systematic ranking benefit. The $K = 1$ configuration achieves the highest Recall@20 in seven of the nine encoder–dataset pairs; the only exceptions are BPRMF on Clothing, where $K = 2$ is marginally higher, and VBPR on Baby, where $K = 2$ and $K = 4$ tie for the best result. Larger ranks can instead reduce ranking performance, most visibly for LightGCN, for which Recall@20 decreases with K on all three datasets. Thus, within the tested configurations, additional monotonic components are not required for strong top-k performance.

Number of squared components Table 6 varies $S \in \{1, 2, 4\}$ while leaving the remaining circuit configuration unchanged. Ranking strongly favours the simplest structure. $S = 1$ attains the highest Recall@20 in eight of the nine encoder–dataset pairs and outperforms the larger-S alternatives in

Table 6: Effect of the number of squared components S (Equation (4)) on NLL and Recall@20. **Bold** denotes the best NLL and Recall within each encoder–dataset pair.

		Baby		Clothing		DBbook	
Encoder	S	NLL↓	R@20↑	NLL↓	R@20↑	NLL↓	R@20↑
BPRMF	1	**8.407**	**0.0890**	**9.734**	**0.0687**	**6.974**	**0.2689**
	2	8.519	0.0820	11.192	0.0605	7.217	0.2549
	4	8.514	0.0787	12.325	0.0544	7.113	0.2512
LightGCN	1	8.241	0.0903	9.507	**0.0702**	**6.811**	**0.2744**
	2	8.165	**0.0916**	9.451	0.0687	6.978	0.2699
	4	**8.038**	0.0916	**9.396**	0.0680	6.896	0.2676
VBPR	1	8.183	**0.0960**	9.241	**0.0985**	6.924	**0.2694**
	2	8.262	0.0926	**9.209**	0.0943	**6.896**	0.2690
	4	**8.090**	0.0953	9.911	0.0925	6.996	0.2593

16 of 18 pairwise comparisons. The only exception is LightGCN on Baby, where $S = 2$ and $S = 4$ attain the same highest displayed Recall@20. Thus, partitioning the representation into additional squared components provides no systematic benefit for top-k recommendation.

Held-out likelihood exhibits a different pattern. Relative to $S = 1$, larger values of S reduce test NLL in four of six LightGCN comparisons and three of six VBPR comparisons, but in none of the six BPRMF comparisons. Additional squared structure can therefore improve the fit of the conditional distribution without improving the ordering of its highest-scoring items.

Joint variation of S and K Table 7 jointly varies the two structural choices. The one-dimensional ablations appear as slices of this grid: the $K = 1$ column recovers the square-count comparison in Table 6, while the $S = 1$ rows recover the monotonic-rank comparison in Table 5.

These results reinforce the distinction between ranking and likelihood. A configuration with $S = 1$ achieves the highest Recall@20 in eight of the nine encoder–dataset blocks; only LightGCN on Baby prefers a larger number of squares. In contrast, the best-NLL configuration uses $S > 1$ in all six LightGCN and VBPR blocks, whereas all three BPRMF blocks favour $S = 1$. The usefulness of additional squared structure for distribution modelling is therefore encoder-dependent, while its contribution to ranking remains limited.

Overall, these experiments indicate that the circuit structure preferred for modelling the full conditional distribution need not be the one preferred for top-k recommendation. Additional squared components can improve held-out likelihood, particularly for LightGCN and VBPR, while the simplest $S = 1$ structure remains sufficient for ranking in most settings.

Table 7: NLL and Recall@20 over the $S \in \{1, 2, 4\}$ and $K \in \{1, 2, 4\}$ circuit-structure grid (see Equation (4)). Within each encoder–dataset–S block, **bold** and underlined values denote the best and second-best results, respectively.

			Baby		Clothing		DBbook	
Encoder	S	K	NLL↓	R@20↑	NLL↓	R@20↑	NLL↓	R@20↑
BPRMF	1	1	8.407	**0.0890**	**9.734**	0.0687	**6.974**	**0.2689**
		2	**8.395**	0.0886	9.748	**0.0688**	6.996	0.2682
		4	8.399	0.0887	9.741	0.0687	6.989	0.2682
	2	1	8.519	0.0820	11.192	0.0605	7.217	0.2549
		2	8.514	0.0815	11.216	**0.0609**	7.081	**0.2530**
		4	**8.512**	**0.0822**	**10.655**	0.0596	**7.058**	0.2538
	4	1	8.514	0.0787	**12.325**	0.0544	**7.113**	**0.2512**
		2	8.513	**0.0782**	12.417	0.0548	7.267	0.2507
		4	**8.506**	0.0785	12.372	**0.0550**	7.270	0.2501
LightGCN	1	1	**8.241**	**0.0903**	**9.507**	**0.0702**	**6.811**	**0.2744**
		2	8.296	0.0858	9.555	0.0679	6.894	0.2731
		4	8.343	0.0831	9.604	0.0682	6.959	0.2663
	2	1	**8.165**	**0.0916**	9.451	**0.0687**	6.978	0.2699
		2	8.239	0.0868	9.477	0.0682	**6.792**	0.2719
		4	8.458	0.0881	9.507	0.0668	6.856	0.2712
	4	1	**8.038**	**0.0916**	**9.396**	0.0680	6.896	**0.2676**
		2	8.136	0.0908	9.429	0.0670	6.984	0.2565
		4	8.436	0.0837	9.471	**0.0682**	**6.779**	0.2651
VBPR	1	1	8.183	0.0960	9.241	**0.0985**	6.924	**0.2694**
		2	**8.027**	**0.0979**	**9.235**	0.0973	6.922	0.2689
		4	8.327	0.0969	9.314	0.0974	**6.915**	0.2684
	2	1	8.262	0.0926	**9.209**	**0.0943**	6.896	**0.2690**
		2	**8.154**	**0.0970**	9.588	0.0937	**6.865**	0.2589
		4	8.258	0.0956	9.589	0.0935	6.874	0.2623
	4	1	**8.090**	0.0953	**9.911**	**0.0925**	6.996	**0.2593**
		2	8.113	0.0951	10.029	0.0919	7.009	0.2545
		4	8.126	**0.0971**	10.018	0.0920	**6.851**	0.2582

Table 8: Per-epoch wall-clock ratio between μSQ and the original recommender, reported as median and range across five datasets. Ratios below one favour μSQ. Bold marks the lower median across widths within each encoder.

Encoder	Median/64	Range/64	Median/256	Range/256
BPRMF	0.63	[0.47, 0.79]	**0.58**	[0.50, 0.69]
LightGCN	**0.42**	[0.41, 0.50]	0.60	[0.58, 0.67]
NGCF	0.57	[0.53, 0.62]	**0.52**	[0.40, 0.64]
VBPR	**0.56**	[0.52, 0.64]	0.68	[0.49, 0.76]
BM3	**2.05**	[1.97, 2.13]	2.78	[2.12, 3.35]
FREEDOM	1.75	[1.60, 2.12]	**1.57**	[1.47, 1.80]
PGL	**1.19**	[1.07, 1.46]	1.26	[1.12, 1.57]
SLMRec	**1.43**	[1.21, 1.61]	2.73	[1.14, 2.92]
SGL	**2.18**	[2.03, 2.36]	2.91	[2.38, 3.46]

D TIME MEASUREMENTS

We compare the wall-clock cost of CiRe with that of each encoder's original training procedure. For every encoder and embedding width, we measure the time required to complete one training epoch for (i) the original recommender with its native objective and (ii) its μSQ circuitisation trained by maximum likelihood. Both measurements include the complete training loop. We report

$$\rho = \frac{\text{seconds per epoch of } \mu\text{SQ}}{\text{seconds per epoch of the original model}},$$

so $\rho < 1$ indicates that the circuitised model is faster. Table 8 reports the median ratio across the five datasets together with its range.

The aggregate measurements reveal a clear dependence on the underlying encoder. For BPRMF, LightGCN, NGCF, and VBPR, μSQ is faster than the original training procedure across the median dataset. In contrast, BM3, FREEDOM, PGL, SLMRec, and especially SGL incur additional cost. To understand this difference, we separately profile the main components of a training step at $d = 64$.

Negative sampling can dominate simple anchors Sampling 1024 negative items takes 8.1–9.9 ms across the five datasets and two embedding widths (median 8.2 ms, $n = 80$). Its cost varies little with catalogue size: the median is 8.2 ms on DBbook and 8.6 ms on Clothing, despite the latter having more than four times as many items. This overhead is substantial for lightweight recommenders. For BPRMF, for example, the complete profiled μSQ step takes 2.3 ms, less than one third of the negative-sampling cost alone. Removing sampled pairwise training can therefore more than compensate for the cost of exact normalisation in such models.

Gram computation is comparatively small The additional operation specific to maximum-likelihood training with μSQ is the construction and contraction of Gram matrices. Across the profiled models, this contribution is approximately 1.1–2.4 ms per batch. Its cost is determined by catalogue size, embedding width, and the required contractions rather than by the recommender architecture itself. Individual Gram construction is highly efficient on the GPU: at $d = 256$, for example, one Gram takes 0.687 ms on DBbook ($I = 5{,}217$) and 0.684 ms on Clothing ($I = 22{,}498$). Thus, in the lightweight models, the cost introduced by exact normalisation is smaller than the sampling overhead it replaces.

Encoder and auxiliary costs govern the computational trade-off Several recommenders optimise auxiliary terms in addition to the principal recommendation objective. In the original implementations, sampled negatives can often be shared across these terms, whereas the corresponding circuitised objectives may require additional normalisation computations. FREEDOM, for example, evaluates three normalisers where its original training procedure samples negatives once. Consequently, when auxiliary computation already accounts for a large fraction of the training step—approximately 45% for BM3, 57% for FREEDOM, and 96.5% for SGL– the saving from removing negative sampling is insufficient to offset the additional circuit computation.

These measurements explain why the computational effect of circuitisation is architecture-dependent. It can reduce training cost when negative sampling represents a substantial fraction of

an otherwise inexpensive training step, while propagation-heavy encoders and auxiliary-objective-heavy models can incur a net overhead.

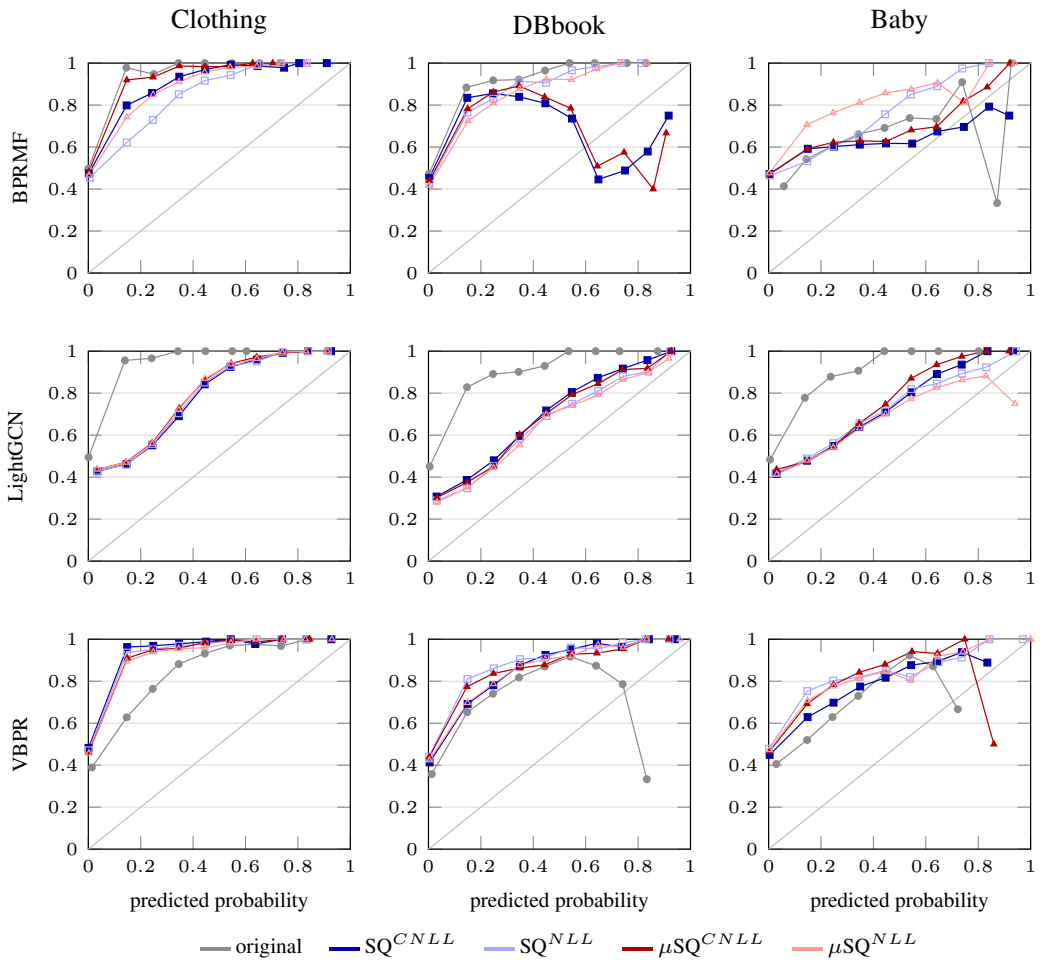

Figure 5: Calibration diagrams of circuits versus base recommendation models: ratio of positives against predicted probability, one panel per encoder and dataset, one line per variant, with the occupied-bin ECE beside each label. Curves closer to the diagonal are better calibrated.

E CALIBRATION

Exact normalisation makes $p_\theta(i \mid u)$ a well-defined categorical distribution, but does not imply by itself that the model's scores are calibrated as binary indicators of observed interactions. Thus, we evaluate a separate binary calibration diagnosticLoconte et al. (2023), whose reliability curves are shown in Figure 5. Each held-out positive interaction is paired with one sampled unobserved item, yielding a balanced binary prediction task. Negatives exclude items already observed by the user and relevant test positives. Original recommenders contribute their raw scores, while SQ and μSQ contribute their log masses, $s(u,i) = \log q_\theta(u,i)$. These scores are mapped to $[0,1]$ using min–max normalisation. Accordingly, this experiment evaluates calibration under the sampled binary interaction task.

Because these quantities are defined on different scales, we map them to $[0,1]$ using training-set min–max normalisation,

$$\widehat{c}_{\mathrm{mm}}(u,i) = \mathrm{clip}_{[0,1]}\left(\frac{s(u,i) - s^{\mathrm{train}}_{\mathrm{min}}}{s^{\mathrm{train}}_{\mathrm{max}} - s^{\mathrm{train}}_{\mathrm{min}}}\right). \tag{16}$$

This provides a common bounded scale without imposing a model-specific link function. Predictions are assigned to ten uniform-width bins and we compute the mean absolute deviation between the

Table 9: Min–max score-based calibration diagnostic. Values are $\mathrm{ECE}_{\mathrm{occ}}$ over ten uniform-width bins; lower is better. **Bold** denotes the lowest value within each encoder–dataset block.

Encoder	Variant	Baby	Clothing	DBbook
BPRMF	Original	0.2759	0.4988	0.4684
	SQ	**0.2062**	**0.4155**	**0.3679**
	μSQ	0.2144	0.5286	0.3872
LightGCN	Original	0.4857	0.5846	0.4486
	SQ	**0.2513**	**0.2943**	0.2118
	μSQ	0.2752	0.3080	**0.1978**
VBPR	Original	**0.3251**	**0.3629**	0.3774
	SQ	0.3342	0.4485	**0.3680**
	μSQ	0.4186	0.4771	0.3741

mean transformed score and empirical positive frequency over occupied bins:

$$\mathrm{ECE}_{\mathrm{occ}} = \frac{1}{|\mathcal{B}_+|} \sum_{b \in \mathcal{B}_+} \left| \frac{1}{|B_b|} \sum_{j \in B_b} \widehat{c}_j - \frac{1}{|B_b|} \sum_{j \in B_b} Y_j \right|, \tag{17}$$

where $\mathcal{B}_+$ denotes the set of nonempty bins. Unlike conventional sample-weighted ECE, each occupied bin contributes equally.

At least one circuit variant achieves lower ECE than its matched original recommender in seven of the nine encoder–dataset pairs. The two exceptions are VBPR on Baby and Clothing. Among the circuit heads, SQ obtains lower ECE than μSQ in eight of nine matched comparisons, with LightGCN on DBbook as the only exception. This ordering differs from the ranking results, where μSQ is often competitive with or stronger than SQ, indicating that top-k effectiveness and this score-based diagnostic capture different properties of the learned interactions.