Table 1: Comparison results with baselines on three datasets w.r.t. four metrics.

	CiteULike				MovieLens				Gowalla			
	P@10	NDCG@10	NDCG@50	MRR	P@10	NDCG@10	NDCG@50	MRR	P@10	NDCG@10	NDCG@50	MRR
SD-GAR	**0.0366**	**0.0997**	**0.1365**	**0.1409**	**0.1311**	**0.2254**	**0.3134**	**0.3711**	**0.0652**	**0.1120**	**0.1620**	**0.2366**
CFGAN	0.0031	0.0091	0.0109	0.0147	0.0471	0.0684	0.1076	0.1425	0.0009	0.0026	0.0031	0.0066
CFGAN*	0.0070	0.0205	0.0296	0.0345	0.0661	0.1130	0.1660	0.2101	0.0207	0.0336	0.0428	0.0873
VAECF	0.0206	0.0519	0.0820	0.0761	0.0764	0.1300	0.2251	0.2214	0.0424	0.0735	0.1172	0.1583
U-GAR	0.0347	0.0897	0.1269	0.1305	0.1148	0.1920	0.2764	0.3251	0.0605	0.1048	0.1490	0.2265
BCE	0.0298	0.0753	0.1071	0.1121	0.0984	0.1650	0.2362	0.2913	0.0504	0.0870	0.1261	0.1927

Q1: It is better to compare the SOTA methods (e.g. CFGAN, VAECF, WRMF, NCF), show results on various metrics.

A1: We perform a comparison with CFGAN and VAECF with source codes released by the original authors. For CFGAN, both the author's suggested setting and our optimally tuned setting (CFGAN*) are tested. For VAECF, the parameters are optimally tuned following the original author's suggestion. (CML is extensively verified to be better than WRMF; while NCF is not a fair comparison for MF-based approaches, as it is an ensemble of MF and NN.) We also report performances on a wide spectrum of extra metrics, like MRR, P@10 and NDCG@10. It can be observed that our approach, SD-GAR, significantly outperforms these additional baselines, which further verifies our effectiveness.

Q2: It is necessary to show more thorough ablation studies such as the effect of the loss function, the effect of different sampling methods, and the effect of training methods.

A2: SD-GAR's advantages can be fully explained with two ablation studies. 1) U-GAR replaces the proposed sampler with uniform sampler, and 2) BCE substitutes the loss function with binary cross entropy. Given the substantial performance gain in Table 1, we may conclude that 1) SD-GAR's sampling strategy is much better than conventional uniform sampling; 2) the proposed loss function is more effective than commonly used BPR (already reported) and BCE. Note that the alternative training method, i.e., policy gradient, is not compared, as it calls for normalization over all items of $\boldsymbol{y}_{\cdot k}$ for each k, which is so time-consuming that training can not be completed within a short rebuttal period with straightforward implementation. The improved performance of SD-GAR already verifies the effectiveness of the proposed training method; we will further study how to implement policy gradient efficiently in future work.

Q3: It is better to state the connection from the proposition 2.1 to the variance of the estimator. In addition, it is better to explain the relation between maximizing Eq. 7 and minimizing estimator variance.

A3: Following asymptotic unbiasedness of the estimator, we show its variance so as to provide guidance for variance reduction, which can help optimize the proposal Q. We will follow reviewer's suggestion to connect them more smoothly. According to Theorem 2.2, with entropy regularization, the optimum of Eq (7) is achieved when $\boldsymbol{x}_c^\top \boldsymbol{y}_i \propto \exp\left(P_{G_T^\star}(i|c)|f_c(i) - \mu_c|/T\right)$, which is approximately $\propto P_{G_T^\star}(i|c)|f_c(i) - \mu_c|$ when T is comparatively large.

Q4: The optimal hyperparameter values are not thoroughly declared for the competitors, e.g., the L2-regularization coefficient in BPR and the margin size in CML.

A4: The parameters for the baselines are optimally tuned within the following scopes. For all the competitors, the L2-regularization coefficient is tuned over {0.01, 0.03, 0.05}. The margin size in CML is tuned over {0.5, 1.0, 1.5, 2.0}. Other hyperparameters, e.g., the embedding size and the number of negative samples are set to the same as SD-GAR.

Q5: In case that the size of the latent dimension is small, comparing Gan-like methods with traditional latent factor models is unfair because the number of parameters in Gan-like methods is much larger than that in latent factor models

A5: In fact, our recommender only uses D, which is of the same size as other latent factor models, for recommendation. While G is only a sampler which is used to help with the training of D, it does not take part in the prediction of recommendation score. Therefore, the comparison is fair for all the reported methods.

Q6: The authors claimed that the discriminator D (rather than the generator G) should be considered as a recommender due to the data sparsity issue in the generator G. However, the authors also repeatedly mentioned that the discriminator D shows poor performances in top-k recommendation, which makes the readers confused.

A6: Generally speaking, *the discriminator (D) rather than generator (G) is more suitable for making recommendation, because D learns directly from training data, whereas G merely learns from samples drawn from the generator distribution; besides, the learning of G is guided by D, which can be not reliable.* Unfortunately, D is not well trained in IRGAN, as G is pretrained, which becomes more likely to generate "hard cases", and in return harms the training performance of D in the initial stage. Our proposed framework does not have such a limitation: the generator is not required to be initiated highly accurate. Instead, the accuracy of G is improved simultaneously with D: when D is initialized, G only shows it with easy cases; when D improves, G will be enhanced as well and gradually present more difficult cases. As a result, G will always contribute to D's training performance without introducing any side effect.

Finally, we will follow reviewers' other suggestions to make discussion on related works, e.g., LightGCN, CFGAN, VAECF, etc, and revise all the typos and unclear expressions.