Out-of-Distribution Detection via Conditional Kernel Independence Model - Supplementary File

Yu Wang[*1] **, Jingjing Zou**[*2]**, Jingyang Lin**[3]**, Qing Ling**[3]**, Yingwei Pan**[4]**, Ting Yao**[4]**, Tao Mei**[4]
[1]: Qiyuan Lab, Beijing, China
[2]: University of California, San Diego, USA
[3]: Sun Yat-sen University, Guangzhou, China
[4]: JD AI Research, Beijing, China
feather1014@gmail.com, j2zou@ucsd.edu, yung.linjy@gmail.com
lingqing556@mail.sysu.edu.cn, {panyw.ustc, tingyao.ustc}@gmail.com, tmei@jd.com

1 Introduction

This supplementary file provides evidences that support the claims submitted in the main paper. We refer to Section/Table/Figure/Eq. in the main paper as Section M.xx/Table M.xx/Figure M.xx/Eq.M.(xx). We denote Section/Table/Figure/Eq. in this supplementary file as Section S.xx/Table S.xx/Figure S.xx/Eq. S.(xx).

In brief summary, the following contents are included: 1. The proof supporting Theorem 1 of the main paper. 2. More explanations behind Section M.4.1 that distinguish Conditional-i from HOOD in principle. 3. How we generated OOD training data for Conditional-i-generative. 4. Algorithm flow for Conditional-i. 5. Implementation details of implementing Conditional-i-generative. 6. More implementation and evaluation details of experiments demonstrated in M.Section 5, including how the error bar was produced, sampling strategy and other setups. 7. HSIC metric comparisons between compared methods. 8. Code Licenses. 9. Detailed performance scores corresponding to each specific test dataset.

2 Proof of Theorem 1

An outline of proof for Theorem 1 is provided below by adapting techniques from [9]. For convenience, let $h^{(c)} = F^{(c)} \circ (g^{(1)}, \ldots, g^{(C)}, g^{(o)})$ denote the mapping from the sample-generating random vectors to the feature of class c extracted from the samples. Therefore for a random sample $(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i)$ of the generating vector $(\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(C)}, \boldsymbol{\beta})$, where $\boldsymbol{w}^{(c)}$ denotes the generator of a class c incidence, $\boldsymbol{\beta}$ denotes the generator of an outlier, and $h^{(c)} : (\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i) \mapsto \boldsymbol{z}_i^{(c)}$, for $c = 1, \ldots, C$. We also denote the lth element of $h^{(c)}$ as $h_l^{(c)}$ for $l = 1, \ldots, d$.

By intermediate value theorem,

$$\left|\frac{\partial F_l^{(c)}}{\partial [\boldsymbol{\beta}_i]_j}\right| \leq \left|h_l^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i + \Delta \cdot e_j) - h_l^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i - \Delta \cdot e_j)\right| / (2\Delta) + |\Delta^2 \phi_{lj}'''/6|, \tag{1}$$

where Δ is a small constant, e_j is the unit vector in the direction of $[\boldsymbol{\beta}_i]_j$, and ϕ_{lj}''' is the third derivative of $h_l^{(c)}$ as a function of $[\boldsymbol{\beta}_i]_j$ alone (fixing values of other inputs) at a value in the interval $([\boldsymbol{\beta}_i]_j - \Delta, [\boldsymbol{\beta}_i]_j + \Delta)$.

[*] Yu Wang and Jingjing Zou contributed equally to this work (joint first author).

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Let

$$\left\| h^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i + \Delta e_j) - \boldsymbol{w}_i^{(c)} \right\|_2^2 = \epsilon_1$$

and

$$\left\| h^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i - \Delta e_j) - \boldsymbol{w}_i^{(c)} \right\|_2^2 = \epsilon_2,$$

then the right side of Eq. S.(1) is bounded further using the triangular inequality by

$$\frac{\sqrt{\epsilon_1} + \sqrt{\epsilon_2}}{2\Delta} + \left| \frac{\Delta^2 \phi_{lj}'''}{6} \right|,$$

and

$$E\left[\left| \frac{\partial F_l^{(c)}}{\partial [\boldsymbol{\beta}_i]_j} \right| \right] \leq \frac{E(\sqrt{\epsilon_1}) + E(\sqrt{\epsilon_2})}{2\Delta} + \left| \frac{\Delta^2 \phi_{lj}'''}{6} \right|.$$

By Jensen's inequality, both $E(\sqrt{\epsilon_1})$ and $E(\sqrt{\epsilon_2})$ are bounded by $[E(\|h^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i) - \boldsymbol{w}_i^{(c)}\|_2^2)]^{1/2}$, which is in turn bounded by the square root of

$$\frac{1}{N} \sum_{i=1}^{N} \|h^{(c)}(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)}, \boldsymbol{\beta}_i) - \boldsymbol{w}_i^{(c)}\|_2^2 + K_1 R_N + 4C_f^2 \sqrt{\log(1/\delta)/(2N)}, \tag{2}$$

where K_1 is a constant and R_N is the Rademacher complexity of function class $F_l^{(c)}(\cdot; \boldsymbol{\theta})$ for all $l = 1, \ldots, d$ and $c = 1, \ldots, C$, based on Theorem 3.3 of [10].

The first term in Eq. S.(2) is assumed to be bounded by $\nu \geq 0$. The rationale for a small ν is that an optimal solution F should minimize Eq. (10) in the main text, which is equivalent to minimizing the empirical distance between the softmax function of $(\boldsymbol{z}_i^{(1)}, \ldots, \boldsymbol{z}_i^{(C)})$ and that of the true classifiers $(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)})$. The empirical distance between the softmax functions is bounded and Lipschitz and can be approximated by the L_2 norm of the distance between $(\boldsymbol{z}_i^{(1)}, \ldots, \boldsymbol{z}_i^{(C)})$ and $(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(C)})$, and thus we expect to see a small empirical distance given a well-performing optimal solution F. Therefore, an upper bound for Eq. S.(2) is given by $\epsilon = \nu + K_1 R_N + 4C_f^2 \sqrt{\log(1/\delta)/(2N)}$. Evaluating the expectation of Eq. S.(1) with conditions in Theorem 1 gives

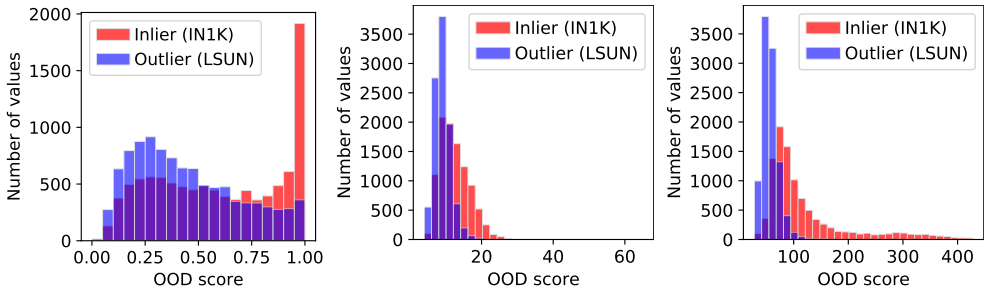
$$E\left[\left| \frac{\partial F_l^{(c)}}{\partial [\boldsymbol{\beta}_i]_j} \right| \right] \leq \frac{\sqrt{\epsilon}}{\Delta} + \frac{\Delta^2 C_d}{6},$$

and finding the minimizer Δ leads to the desired result.

3 Examples Supporting Section M.4.1

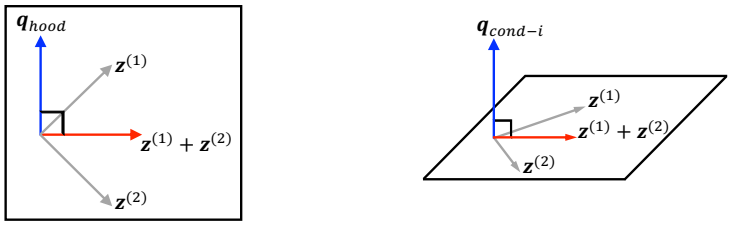
Here we provide an example to demonstrate the difference in solution between the Conditional-i and HOOD [7] methods. With results of Sections 3.4 and 4.1 in the main text, in the scenario of $d = 1$ and linear kernel, $\text{HSIC}_{\text{HOOD}} = \|\boldsymbol{Z}^\top \boldsymbol{Q}\|_F$, where $\boldsymbol{Z}$ denotes the vector of samples from the mixture distribution $p_M(\boldsymbol{z}) = \sum_{c=1}^{C} p_\theta(\boldsymbol{z}^{(\xi)} \mid \xi = c) p(\xi = c)$ and $\boldsymbol{Q}$ denotes the samples of out of distribution features $\boldsymbol{q}$. When the sample size is large, $\text{HSIC}_{\text{HOOD}}$ is approximately the Frobenius norm of N times of the correlation between the out of distribution feature and $\boldsymbol{z}$ from the mixture distribution. Minimizers $\boldsymbol{q}$ of $\text{HSIC}_{\text{HOOD}}$ are those orthogonal (inner product defined by the correlation) to the linear combination $\sum_{c=1}^{C} p(\xi = c) \boldsymbol{z}^{(c)}$, where $\boldsymbol{z}^{(c)}$ follows the conditional distribution of $\boldsymbol{z}$ given class $\xi = c$. Suppose the total number of classes $C = 2$, $p(\xi = 1) = p(\xi = 2) = 1/2$, and the correlation between $z^{(1)}$ and $z^{(2)}$ is omissible. Then $\boldsymbol{q} = \boldsymbol{z}^{(1)}/\sqrt{2} - \boldsymbol{z}^{(2)}/\sqrt{2}$ minimizes $\text{HSIC}_{\text{HOOD}}$ but is not orthogonal of each $\boldsymbol{z}^{(c)}$. On contrary, the minimizer of $\text{HSIC}_{\text{cond-i}}$ is required to be orthogonal of each $\boldsymbol{z}_i^{(c)}$.

We also associate new visualization results to support the difference between HOOD and Conditional-i. Figure S.1 demonstrates how well the test score functions out of each method separates the in-distribution and out-of-distribution data. It can be observed that Conditional-i (Figure S.1(c)) offers better separation than HOOD (Figure S.1(b)), as per the discussion in the response. HOOD



(a) K-Base (AUROC: 65.36) (b) HOOD (AUROC: 79.62) (c) Conditional-i (AUROC: 85.20)

Figure 1: OOD test score histograms out of different OOD detection methods.



(a) Minimizer $\boldsymbol{q}_{hood}$ of HOOD (b) Minimizer $\boldsymbol{q}_{cond-i}$ of Conditional-i

Figure 2: Illustration of HOOD optimum and Conditional-i optimum

and Conditional-i both significantly surpass the K-Base model (Figure S.1(a)) in terms of test score separation. The features used here were obtained from the model producing Table M.2 in the main paper.

In terms of the illustrative example where $d = 3$ we mentioned during response, we also managed to illustrate our example in Figure S.2.

Specifically, in the scenario of $d = 3$ and linear kernel, $\text{HSIC}_{\text{hood}} = |\boldsymbol{Z}^\top \boldsymbol{Q}|_F$. We assume, for the purpose of visual illustration, in-distribution feature from the 1st in-distribution class $\boldsymbol{z}^{(1)}$ is virtually represented by the vector $[1, 0, 0]^\top \in R^3$, and the 2nd in-distribution class $\boldsymbol{z}^{(2)}$ is virtually represented by $[0, 1, 0]^\top \in R^3$. Suppose again the total number of classes $C = 2$, $p(\xi = 1) = p(\xi = 2) = 1/2$, and the correlation between $z^{(1)}$ and $z^{(2)}$ is omissible, made apparent by the orthogonal relation between $z^{(1)}$ and $z^{(2)}$. Then OOD feature $\boldsymbol{q} = [-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0]$ is one of the minimizers for $\text{HSIC}_{\text{hood}}$ but is not orthogonal of each individual $\boldsymbol{z}^{(1)}$ or $\boldsymbol{z}^{(2)}$ from in-distribution classes at all. This means that $\boldsymbol{q}$ still poses strong correlations/dependence with both in-distribution features $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$, whereas $\boldsymbol{q}$ is independent of the standardized sum $[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0]$ of the two (HOOD motivation), which represents the mixture distribution of the two features.

However, the HOOD minimizer $\boldsymbol{q} = [-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0]$ cannot be a minimizer of $\text{HSIC}_{\text{cond-i}}$, as Conditional-i poses stricter conditions that $\boldsymbol{q}$ needs to be having small dependence on every in-distribution class. Correspondingly, $\boldsymbol{q} = [0, 0, 1]$ is an optimal solution of Conditional-i loss, that $\boldsymbol{q} = [0, 0, 1]$ achieves simultaneous independence with both $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$. Note that the desired solution $\boldsymbol{q} = [0, 0, 1]$ can also be a local minimum of HOOD though, but HOOD has no incentive to avoid the solution of $\boldsymbol{q} = [-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0]$ among infinite number of local minima, whereas Conditional-i exclusively can escape such bad local minima, by always anchoring at the desired solution $\boldsymbol{q} = [0, 0, 1]$.

This example is well illustrated by Figure S.2, where $\boldsymbol{q}_{hood}$ illustrates the minimizer of HOOD whereas $\boldsymbol{q}_{cond-i}$ represents the minimizer of Conditional-i. Minimizer of OOD feature $\boldsymbol{q}_{hood}$ out of HOOD lies in the same hyperplane spanned by $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$. Note that $\boldsymbol{q}_{hood}$ is orthogonal to

the sum of $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$, whereas $\boldsymbol{q}_{hood}$ is still correlated to both $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$. Minimizer of OOD feature $\boldsymbol{q}_{cond-i}$ out of Conditional-i lies in the position orthogonal to the hyperplane spanned by $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$, and $\boldsymbol{q}_{cond-i}$ is orthogonal to both of $\boldsymbol{z}^{(1)}$ and $\boldsymbol{z}^{(2)}$.

4 Overall Algorithm Flow of Conditional-i

```
# encoder: f;  cls: classifier;
# x, y: inlier images and labels;
# u: outlier images; C: number of inlier classes;
# C_hat: number of classes for CondHSIC;
# lamb: conditional-i loss weight; queue: inlier queue;

for (x, y, u) in loader:
  # encoding features
  z, q = f(x), f(u)  # (N, D)
  # cross entropy loss
  p = cls(z). # (N, C)
  ce_loss = CrossEntropyLoss(p, y)
  # conditional-i loss
  cond_hsic_loss = CondHSIC(q, queue).mean()
  loss = ce_loss + lamb * cond_hsic_loss # total loss
  loss.backward()
  # update encode and classifier
  update(f, cls)
  # update inlier queue based on class of z
  enqueue_n_dequeue(queue, z, y)

def CondHSIC(q, queue):
  C = queue.size(dim=0) # number of inlier classes
  # random select a list of inlier classes
  C_hat_list = randint(C, size=C_hat)
  # conditional hsic loss
  cond_hsic = 0.
  for i in C_hat_list:
    cond_hsic += HSIC(q, queue[i]) / C_hat
  return cond_hsic

def HSIC(q, z):
  N = z.size(1)
  # compute batch-wise kernel K
  K_z, K_q = b_kernel(z), b_kernel(q)  # (C, N, N)
  # compute KH
  KH_z = K_z - mean(K_z, dim=1, keepdim=True)
  KH_q = K_q - mean(K_q, dim=1, keepdim=True)
  return trace(bmm(KH_z, KH_q))/(N-1)**2
```

Figure 3: PyTorch pseudo-code of Conditional-i Model.

5 Conditional-i-generative

In this section, we describe how we did generate "fake" OOD training data for Conditional-i-generative, particularly for the training scenarios when we lose access to true known OOD training data. Briefly speaking, we apply distortions to each of the inlier data so that we can generate OOD training data based on in-distribution data themselves through relatively cheap implementations.

In order to produce the Conditional-i-generative scores reported in Table M.1, we apply strong augmentations [2] technique to each inlier training image with the augmentation strength $N = 5$, $M = 30$ [2]. The strong augmentation then effectively applies strong distortions to inlier data, leading to potential distributional shift of the inlier features [11]. We treat these distorted inlier data

as training OOD data which was then used to compute the outlier kernel matrix Φ_q of Conditional-i during training.

In order to produce Conditional-i-generative scores reported in Table M.2, we apply both strong augmentations [2] technique and CutMix techniques to generate the fake OOD training data. We firstly apply strong augmentation with the augmentation strength $N = 5$, $M = 30$ [2] on top of the inlier training image. We then apply the CutMix [13] technique between each random paring of inlier data out of the strong augmentation approach. Specifically, we employ each inlier data as the reference image, and we randomly cut patches from another inlier data. We apply the CutMix approach to mix the patch with the reference image as described in [13]. This procedure also effectively helps us to generate desired feature level distributional shift [11] to generate OOD data. For the hyperparameters used in CutMix, we set $\alpha = 0.2$. The resultant feature is then used as training outliers that forms the outlier matrix Φ_q. This procedure also corresponds to the method "+RandAug+Cutmix" described in Table M. 7.

6 Implementation Details of Section M.5

This section describes supplementary implementation details of the experiments reported in Section M.5. The reported details include: hyperparameters for training baseline models and Conditional-i models in the main paper; Ablations against the τ values for Conditional-i, showing that Conditional-i is relatively robust against the choice of kernel temperature τ; Error bar calculation rules; More optimization details complementary to the main file; The complete OOD detection test scores across each specific test datasets, based on which we obtain the averaged score Table M.1, M.2, M.3 in the main file.

6.1 Hyperparameters for other baseline models compared in Section M.5

For all compared methods, we believe we have tuned each of the methods to our best by searching the optimal hyperparameters in the suggested searching range. The optimal hyperparameters (either the default value in the original publication or the optimal value we found) for each method were listed in Table S.1. Other hyperparameters that are not reported in Table S.1 remain as the default values as defined in the original publications. Please see the references in Table S.1 for specific hyperparameter definitions.

Table 1: Optimal hyperparameter values found for Table M.1, M.2 and M.3.

Methods	Table M.1	Table M.2	Table M.3
ODIN [6]	$T = 10^{+3}$, $\epsilon = 0.0$	$T = 10^{+3}$, $\epsilon = 0.0014$	$T = 10^{+4}$, $\epsilon = 0.0$
Mahalanobis [5]	$\epsilon = 0.0014$	$\epsilon = 0.005$	$\epsilon = 0.00005$
ReAct [12]	$p = 99$	$p = 99$	$p = 99.9$
OE [4]	$\lambda = 0.5$	$\lambda = 0.5$	$\lambda = 6.0$
Energy [8]	$m_{in} = -15$, $m_{out} = -5$	$m_{in} = -15$, $m_{out} = -5$	$m_{in} = -15$, $m_{out} = -7$

6.2 More hyperparameters defined for Section M.5

We use Gaussian kernel to train for Conditional-i model. The τ values we used for the Gaussian kernel at training time are illustrated in the following Table S.2.

Table 2: τ and λ values for Table M.1, M.2 and M.3

Method	Table M.1	Table M.2	Table M.3
Conditional-i-generative	$\tau = 5.0$, $\lambda = 0.5$	$\tau = 20.0$, $\lambda = 300.0$	/
Conditional-i	$\tau = 5.0$, $\lambda = 0.03$	$\tau = 4.0$, $\lambda = 0.06$	$\tau = 3.0$, $\lambda = 0.2$

6.3 Ablations against the τ values

We display ablation studies against the temperature τ values under the same training setup used for producing Table M.1. We vary the value of τ and the optimal τ is found to be $\tau = 5.0$. Table S.3 shows that the OOD detection performance is relatively robust against mild changes of τ.

Table 3: Ablations against the τ values

τ values	3.0	4.0	5.0	6.0
FPR95 ↓	35.43	34.40	**33.18**	34.07
AUROC ↑	88.99	89.86	**90.03**	89.81
AUPR ↑	60.84	61.18	**61.48**	61.12

6.4 Ablations against the kernel during training

We have implemented Gaussian kernel, linear kernel and and inverse multiquadric (IMQ) kernels during Conditional-i's training. We empirically found that Gaussian kernel provided the relatively better performance, while other training kernels returns comparable results.

6.5 Error bar calculation

This section describes how we calculated the error bar present in Table M.1 M.2 and M.3. We train each OOD detection approach for 3 training runs and we report the test score averaged across these 3 total training runs. For all training runs, we include the same sampled training data across all the training approaches. To produce the test scores, we test each network out of a single training run by averaging across 10 different test stage inference runs [4]. Each of the error bar produced in our main paper was then reported by averaging across 3 training runs $\times$ 10 test runs in total. The randomness and score variance might have originated from 3 sources: firstly, different training seeds might have affected the network initialization; secondly, the seed have also changed the ordering of training batches; Finally, each 1 out of the 10 tests corresponds to a different partition of the complete test set, while the inlier:outlier test data ratio remains to be a constant ratio 1:5 [4].

6.6 Sampling scheme for training and testing

For every single training/test run across all the compared methods, we ensure that all methods were fed with the same set of sampled training data and test data to ensure strict apple-to-apple comparisons. To reach this goal, we shuffle all the training inliers and training outliers to remove the correlation between the image ID ordering and their classes. Then we fix the data permutation and we sample certain amounts of training OOD data to ensure our training procedure is reproducible. We train each model with the same set of training data for 3 times. The test scores of each method are then averaged across such 3 training models (each model 10 test runs), based on which we report the final averaged performance score. This sampling scheme is different from that being used in HOOD paper [7].

6.7 More optimization details for training NLP task present in Table M.3

For NLP experiments that return the score in Table M.3, we train a 2-layer GRUs [1] for 5 epochs. We adopt Adam optimizer with learning rate $lr = 0.01$ and $batchsize = 64$ to train for the models.

7 HSIC metric comparison between OOD detection methods

We employ Eq. M.(5) compute the HSIC metric of the deep models out of every compared method at the last training epoch (under the same training setup as we produce Table M.1). We then compute the HSIC values of the compared methods using CIFAR-10 test data and OOD test data respectively as $\Phi_z^{(c)}$ and Φ_q. The HSIC and evaluation metrics are illustrated in Table S.4 along with their test accuracy. As shown in Table S.4, Conditional-i training achieves the lowest HSIC metric and highest accuracy among all methods no matter whether we use linear or Gaussian kernel to evaluate the HSIC metric.

Table 4: HSIC metric comparison between OOD detection methods

Metrics Values	HSIC loss Gaussian kernel ($\tau = 5$)	HSIC loss linear kernel	FPR95 ↓	AUROC ↑	AUPR ↑
K-Base	0.0053	16.2912	65.84	74.76	33.25
OE	0.0019	4.2954	33.41	89.75	57.74
Conditional-i	0.0003	0.4850	32.66	90.03	61.48

8 Code Licenses

We downloaded the published code from each baseline method's official website and reimplemented every method. We claim that we only use these code for academic purposes. We also included our code for Conditional-i at https://github.com/OODHSIC/conditional-i. Please kindly do not distribute the code URLs before our paper is eventually published. We promise to release our code upon the paper publication.

9 Complete OOD detection performance over each specific test dataset

This section reports the complete OOD detection performance over each specific test dataset. Please see Table S.5, S.6, S.7 presented below that display the detailed test scores respectively for Table M.1, M.2, M.3.

Table 5: OOD Detection performance of CIFAR-100 as in-distribution for each specific OOD test dataset. Averaged test scores across different test sets are reported in Table M.1 of the main paper.

Dataset $\mathcal{D}_{\text{ood}}^{\text{tst}}$	Methods	True Out. Expo.	FPR95 (%) ↓	AUROC (%) ↑	AUPR (%) ↑
Texture	K-Base [3]	no	63.32	77.06	35.25
	ODIN [6]		65.05	77.82	36.76
	Mahalanobis [5]		**37.88**	**90.86**	**73.32**
	Energy [8]		65.13	77.81	34.50
	ReAct (+Energy) [12]		54.28	80.92	37.20
	Conditional-i-generative (ours)		57.73	89.53	68.32
	Energy [8]	yes	32.08	90.80	56.87
	OE [4]		**30.38**	91.28	61.07
	HOOD [7]		31.25	91.82	68.89
	HOOD-bank		31.55	91.90	72.12
	Conditional-i (ours)		30.71	**92.15**	**72.15**
SVHN	K-Base [3]	no	52.19	80.85	39.94
	ODIN [6]		50.68	82.68	42.53
	Mahalanobis [5]		42.24	86.69	54.07
	Energy [8]		50.13	82.89	39.85
	ReAct (+Energy) [12]		40.04	85.33	41.88
	Conditional-i-generative (ours)		**0.31**	**99.70**	**95.66**
	Energy [8]	yes	**13.79**	94.86	66.55
	OE [4]		18.23	95.05	**73.91**
	HOOD [7]		15.19	**95.29**	70.52
	HOOD-bank		15.19	94.67	66.60
	Conditional-i (ours)		14.61	94.84	71.41
Places365	K-Base [3]	no	62.78	78.83	38.40
	ODIN [6]		66.52	79.02	39.23
	Mahalanobis [5]		62.94	81.28	45.56
	Energy [8]		66.14	78.95	36.35
	ReAct (+Energy) [12]		66.13	78.96	36.40
	Conditional-i-generative (ours)		**45.05**	**89.85**	**61.26**
	Energy [8]	yes	35.11	90.20	59.00
	OE [4]		36.12	89.68	57.90
	HOOD [7]		34.53	90.06	58.17
	HOOD-bank		33.64	90.55	61.38
	Conditional-i (ours)		**33.55**	**90.59**	**61.77**
LSUN	K-Base [3]	no	69.51	75.17	32.24
	ODIN [6]		74.59	74.23	31.56
	Mahalanobis [5]		**62.77**	77.32	35.00
	Energy [8]		74.27	73.85	28.52
	ReAct (+Energy) [12]		77.01	73.55	28.55
	Conditional-i-generative (ours)		65.13	**82.18**	**42.81**
	Energy [8]	yes	**36.13**	89.20	56.99
	OE [4]		39.95	89.11	58.01
	HOOD [7]		39.95	88.47	54.79
	HOOD-bank		37.58	89.13	57.69
	Conditional-i (ours)		36.95	**89.39**	**58.52**
CIFAR10	K-Base [3]	no	**54.81**	80.72	40.56
	ODIN [6]		56.36	81.57	**42.23**
	Mahalanobis [5]		63.04	77.43	33.98
	Energy [8]		55.57	**81.68**	39.73
	ReAct (+Energy) [12]		59.41	80.74	39.11
	Conditional-i-generative (ours)		87.23	57.42	15.38
	Energy [8]	yes	48.22	82.76	**45.17**
	OE [4]		**42.37**	**83.62**	37.82
	HOOD [7]		45.06	83.39	43.51
	HOOD-bank		46.91	83.07	43.02
	Conditional-i (ours)		47.47	83.17	43.57

Table 6: OOD Detection performance of IN1K as in-distribution for each specific OOD test dataset. Averaged test scores across different test sets are reported in Table M.2 of the main paper.

Dataset $\mathcal{D}_{\text{ood}}^{\text{tst}}$	Methods	True Out. Expo.	FPR95 (%) ↓	AUROC (%) ↑	AUPR (%) ↑
Texture	K-Base [3]	no	77.36	67.00	15.22
	ODIN [6]		**61.24**	77.04	23.62
	Mahalanobis [5]		80.14	70.24	**25.88**
	Energy [8]		65.10	74.00	20.57
	ReAct (+Energy) [12]		63.08	76.25	23.94
	Conditional-i-generative (ours)		62.17	**78.01**	24.93
	Energy [8]	yes	**52.38**	**86.82**	**48.21**
	OE [4]		66.57	80.86	38.46
	HOOD [7]		55.56	82.52	39.53
	HOOD-bank		59.09	83.37	42.60
	Conditional-i (ours)		57.34	83.92	44.15
SVHN	K-Base [3]	no	31.27	91.20	65.65
	ODIN [6]		9.32	98.23	**93.12**
	Mahalanobis [5]		35.77	85.51	41.17
	Energy [8]		8.79	98.25	92.57
	ReAct (+Energy) [12]		9.36	98.06	91.91
	Conditional-i-generative (ours)		**4.63**	**98.27**	87.02
	Energy [8]	yes	37.58	83.44	35.87
	OE [4]		25.42	92.58	65.96
	HOOD [7]		10.15	97.39	83.54
	HOOD-bank		20.96	92.47	57.27
	Conditional-i (ours)		**10.00**	**97.41**	**84.01**
Places365	K-Base [3]	no	77.86	68.37	26.88
	ODIN [6]		67.53	75.94	34.16
	Mahalanobis [5]		93.04	46.61	14.77
	Energy [8]		70.90	73.72	30.60
	ReAct (+Energy) [12]		71.92	73.68	30.83
	Conditional-i-generative (ours)		**63.48**	**78.04**	**41.29**
	Energy [8]	yes	60.02	80.49	**44.87**
	OE [4]		70.03	75.80	37.92
	HOOD [7]		62.61	77.58	38.74
	HOOD-bank		60.22	79.58	41.92
	Conditional-i (ours)		**56.84**	**80.61**	42.59
LSUN	K-Base [3]	no	77.74	65.36	23.97
	ODIN [6]		70.62	70.45	**27.23**
	Mahalanobis [5]		94.23	42.04	13.34
	Energy [8]		71.82	68.93	25.70
	ReAct (+Energy) [12]		74.01	67.85	25.06
	Conditional-i-generative (ours)		**70.52**	**70.47**	24.87
	Energy [8]	yes	**45.63**	**87.50**	**57.68**
	OE [4]		67.26	76.64	39.13
	HOOD [7]		56.89	79.62	40.18
	HOOD-bank		53.39	83.27	47.46
	Conditional-i (ours)		47.18	85.20	49.27

Table 7: OOD Detection performance of 20 Newsgroups as in-distribution for each specific OOD test dataset. Averaged test scores across different test sets are reported in Table M.3 of the main paper.

Dataset $\mathcal{D}_{\text{ood}}^{\text{tst}}$	Methods	True Out. Expo.	FPR95 (%) ↓	AUROC (%) ↑	AUPR (%) ↑
SNLI	K-Base [3]		46.21	88.98	71.28
	ODIN [6]		43.82	90.54	76.79
	Mahalanobis [5]	no	90.91	36.06	12.28
	Energy [8]		33.34	93.72	83.94
	ReAct (+Energy) [12]		32.87	93.89	84.47
	Energy [8]		1.08	99.68	98.27
	OE [4]		1.70	99.13	96.92
	HOOD [7]	yes	0.87	99.73	99.29
	HOOD-bank		1.05	99.70	99.04
	Conditional-i (ours)		**0.80**	**99.77**	**99.31**
IMDB	K-Base [3]		50.28	82.43	46.84
	ODIN [6]		41.20	86.44	56.74
	Mahalanobis [5]	no	76.89	60.42	19.56
	Energy [8]		30.62	91.21	64.72
	ReAct (+Energy) [12]		30.82	91.18	64.51
	Energy [8]		2.57	99.18	94.53
	OE [4]		3.05	98.82	93.72
	HOOD [7]	yes	1.69	99.60	97.33
	HOOD-bank		1.74	99.50	97.30
	Conditional-i (ours)		**1.32**	**99.62**	**98.08**
Multi30K	K-Base [3]		65.70	80.50	48.01
	ODIN [6]		56.57	83.85	55.91
	Mahalanobis [5]	no	61.94	72.58	27.43
	Energy [8]		56.96	86.76	61.62
	ReAct (+Energy) [12]		54.69	87.44	62.91
	Energy [8]		0.72	99.68	98.25
	OE [4]		0.77	99.55	98.93
	HOOD [7]	yes	0.17	99.93	99.73
	HOOD-bank		0.25	99.82	99.49
	Conditional-i (ours)		**0.03**	**99.95**	**99.76**
WMT16	K-Base [3]		49.65	85.12	56.24
	ODIN [6]		43.76	87.73	64.59
	Mahalanobis [5]	no	67.64	71.35	27.43
	Energy [8]		39.31	90.27	68.81
	ReAct (+Energy) [12]		38.48	90.62	69.85
	Energy [8]		0.71	99.76	98.50
	OE [4]		0.57	99.86	99.44
	HOOD [7]	yes	0.23	99.94	99.73
	HOOD-bank		0.22	99.95	99.79
	Conditional-i (ours)		**0.05**	**99.97**	**99.80**
Yelp	K-Base [3]		55.92	79.57	40.69
	ODIN [6]		46.81	83.60	48.08
	Mahalanobis [5]	no	56.23	77.33	32.09
	Energy [8]		39.57	87.08	50.62
	ReAct (+Energy) [12]		38.96	87.48	51.81
	Energy [8]		17.48	95.99	82.71
	OE [4]		18.71	95.85	77.45
	HOOD [7]	yes	13.57	95.96	72.50
	HOOD-bank		11.96	96.11	73.06
	Conditional-i (ours)		**10.50**	**96.77**	**76.49**
EWT-A	K-Base [3]		44.63	87.05	33.17
	ODIN [6]		38.27	89.29	44.55
	Mahalanobis [5]	no	72.27	67.23	7.72
	Energy [8]		30.27	92.33	47.24
	ReAct (+Energy) [12]		30.48	92.34	47.28
	Energy [8]		1.03	99.63	93.66
	OE [4]		1.04	99.69	96.54
	HOOD [7]	yes	0.85	99.87	98.01
	HOOD-bank		**0.55**	99.90	**98.54**
	Conditional-i (ours)		**0.55**	**99.91**	98.50
EWT-E	K-Base [3]		39.83	88.95	42.78
	ODIN [6]		36.47	90.21	55.12
	Mahalanobis [5]	no	73.97	60.28	8.32
	Energy [8]		28.71	93.40	59.16
	ReAct (+Energy) [12]		27.80	93.68	60.20
	Energy [8]		0.90	99.71	95.71
	OE [4]		0.88	99.73	97.48
	HOOD [7]	yes	0.50	99.91	98.84
	HOOD-bank		0.51	99.92	98.95
	Conditional-i (ours)		**0.38**	**99.92**	**99.01**
EWT-N	K-Base [3]		71.43	82.17	24.11
	ODIN [6]		66.34	85.28	40.40
	Mahalanobis [5]	no	73.86	65.74	5.52
	Energy [8]		64.91	86.99	37.50
	ReAct (+Energy) [12]		63.98	87.18	38.22
	Energy [8]		1.27	99.62	90.54
	OE [4]		1.26	99.60	94.41
	HOOD [7]	yes	0.97	99.81	95.94
	HOOD-bank		0.86	99.80	96.68
	Conditional-i (ours)		**0.69**	**99.87**	**97.15**
EWT-R	K-Base [3]		49.88	85.06	33.76
	ODIN [6]		42.86	86.95	41.37
	Mahalanobis [5]	no	67.49	69.35	10.89
	Energy [8]		35.55	90.77	48.15
	ReAct (+Energy) [12]		33.66	91.00	48.23
	Energy [8]		1.07	99.57	94.79
	OE [4]		0.93	99.56	97.55
	HOOD [7]	yes	0.47	99.90	98.92
	HOOD-bank		0.45	99.81	98.88
	Conditional-i (ours)		**0.29**	**99.93**	**99.14**
EWT-W	K-Base [3]		54.91	82.89	17.23
	ODIN [6]		50.99	85.14	26.37
	Mahalanobis [5]	no	59.69	73.41	5.77
	Energy [8]		50.76	86.93	24.69
	ReAct (+Energy) [12]		50.71	87.28	25.30
	Energy [8]		1.04	99.72	89.74
	OE [4]		1.03	99.78	95.00
	HOOD [7]	yes	0.95	99.85	95.98
	HOOD-bank		0.70	99.88	97.02
	Conditional-i (ours)		**0.60**	**99.90**	**97.26**

References

[1] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[2] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

[3] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

[4] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

[5] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[6] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[7] Jingyang Lin, Yu Wang, Qi Cai, Yingwei Pan, Ting Yao, Hongyang Chao, and Tao Mei. Out-of-distribution detection with hilbert-schmidt independence optimization. *arXiv:2209.12807*, 2022.

[8] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

[9] Qi Lyu, Xiao Fu, Weiran Wang, and Songtao Lu. Understanding latent correlation-based multiview learning and self-supervision: An identifiability perspective. In *ICLR*, 2022.

[10] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. *MIT press*, 2012.

[11] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. In *ICLR*, 2021.

[12] Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021.

[13] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.