Roadmap. The supplementary material is organized as follows. The details of LogME measurement for object detection are described in A. More details of experiment up in this work are described in Section B. Further, we provide more experimental results regarding the ranking of pre-trained detectors in Section C.

A. LogME Measurement for Object Detection

In this work, we extend a classification assessment method LogME [14] to object detection. In this section, we will give detailed derivations of LogME for object detection framework.

Different from image-level features used for assessing classification task, we extract object-level features of ground-truth bounding boxes by using pre-trained detectors' backbone followed by an ROIAlign layer [9]. In this way, for a given pre-trained detector and a downstream task, we can collect the object-level features of downstream task by using the detector and form a feature matrix $\boldsymbol{F}$, with each row $\boldsymbol{f}_i$ denotes an object-level feature vector. For each $\boldsymbol{f}_i$, we also collect its 4-d coordinates of grounding-truth bounding box $\boldsymbol{b}_i$ and class label c_i to form a bounding box matrix $\boldsymbol{B}$ and a class label matrix $\boldsymbol{C}$.

For the bounding box regression sub-task, LogME measures the transferability by using the maximum evidence $p(\boldsymbol{B}|\boldsymbol{F}){=}\int p(\boldsymbol{\theta}|\alpha)p(\boldsymbol{B}|\boldsymbol{F},\beta,\boldsymbol{\theta})d\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the parameter of linear model. α denotes the parameter of prior distribution of $\boldsymbol{\theta}$, and β denotes the parameter of posterior distribution of each observation $p(\boldsymbol{b}_i|\boldsymbol{f}_i,\beta,\boldsymbol{\theta})$. By using the evidence theory [5] and basic principles in graphical models [6], the evidence can be calculated as

$$\begin{aligned} p(\boldsymbol{B}\,|\,\boldsymbol{F}) &= \int p(\boldsymbol{\theta}|\alpha)p(\boldsymbol{B}|\boldsymbol{F},\beta,\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \int p(\boldsymbol{\theta}\,|\,\alpha)\prod_{i=1}^{M} p(\boldsymbol{b}_i|\boldsymbol{f}_i,\beta,\boldsymbol{\theta})d\boldsymbol{\theta} \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}}\left(\frac{\alpha}{2\pi}\right)^{\frac{D}{2}}\int e^{-\frac{\alpha}{2}\boldsymbol{\theta}^T\boldsymbol{\theta}-\frac{\beta}{2}\|\boldsymbol{f}_i\boldsymbol{\theta}-\boldsymbol{b}_i\|^2}\,\mathrm{d}\boldsymbol{\theta}, \end{aligned} \tag{1}$$

where M is the number of objects and D is the dimension of object features. When A is positive definite,

$$\int e^{-\frac{1}{2}(\boldsymbol{\theta}^T A\boldsymbol{\theta}+b^T\boldsymbol{\theta}+c)}\mathrm{d}\boldsymbol{\theta} = \frac{1}{2}\sqrt{\frac{(2\pi)^D}{|A|}}e^{\frac{1}{4}b^T A^{-1}b-c}. \tag{2}$$

LogME takes the logarithm of Eq. (1) for simpler calculation. So the transferability score is expressed by

$$\begin{aligned} \mathrm{LogME} &= \log p(\boldsymbol{B}|\boldsymbol{F}) \\ &= \frac{M}{2}\log\beta + \frac{D}{2}\log\alpha - \frac{M}{2}\log 2\pi \\ &\quad - \frac{\beta}{2}\|\boldsymbol{F}\boldsymbol{m}-\boldsymbol{B}\|_2^2 - \frac{\alpha}{2}\boldsymbol{m}^T\boldsymbol{m} - \frac{1}{2}\log|A|. \end{aligned} \tag{3}$$

where A and m are

$$A = \alpha I + \beta\boldsymbol{F}^T\boldsymbol{F}, \boldsymbol{m} = \beta A^{-1}\boldsymbol{F}^T\boldsymbol{B}, \tag{4}$$

where A is the L_2-norm of $\boldsymbol{F}$, and $\boldsymbol{m}$ is the solution of $\boldsymbol{\theta}$. Here α and β are maxmized by alternating between evaluating $\boldsymbol{m}$, γ and maximizing α, β with $\boldsymbol{m}$, γ fixed [4] as the following:

$$\gamma = \sum_{i=1}^{D}\frac{\beta\sigma_i}{\alpha+\beta\sigma_i}, \alpha \leftarrow \frac{\gamma}{\boldsymbol{m}^T\boldsymbol{m}}, \beta \leftarrow \frac{M-\gamma}{\|\boldsymbol{F}\boldsymbol{m}-\boldsymbol{B}\|_2^2}, \tag{5}$$

where σ_i's are singular values of $\boldsymbol{F}^T\boldsymbol{F}$. With the optimal α^* and β^*, the logarithm maximum evidence $\mathcal{L}\left(\alpha^*,\beta^*\right)$ is used for evaluating the transferability. Considering $\mathcal{L}\left(\alpha^*,\beta^*\right)$ scales linearly with the number of objects M, it is normalized as $\frac{\mathcal{L}(\alpha^*,\beta^*)}{M}$, which is interpreted as the average logarithm maximum evidence of all given object feature matrix $\boldsymbol{F}$ and bounding box matrix $\boldsymbol{B}$. LogME for classification sub-task can be computed by replacing $\boldsymbol{B}$ in Eq. (3) with converted one-hot class label matrix.

Nevertheless, optimizing LogME by Eq. (4) and Eq. (5) is timely costly, which is comparable with brute-force fine-tuning. So LogME further improves the computation efficiency as follows. The most expensive steps in Eq. (4) are to calculate the inverse matrix A^{-1} and matrix multiplication $A^{-1}\boldsymbol{F}^T$, which can be avoided by decomposing $\boldsymbol{F}^T\boldsymbol{F}$. The decomposition is taken by $\boldsymbol{F}^T\boldsymbol{F} = V\,\mathrm{diag}\{\sigma\}V^T$, where V is an orthogonal matrix. By taking $\Lambda = \mathrm{diag}\{(\alpha+\beta\sigma)\}$, A and A^{-1} turn to $A = \alpha I + \beta\boldsymbol{F}^T\boldsymbol{F} = V\Lambda V^T$ and $A^{-1} = V\Lambda^{-1}V^T$. With associate law, LogME takes a fast computation by $A^{-1}\boldsymbol{F}^T\boldsymbol{B} = \left(V\left(\Lambda^{-1}\left(V^T\left(\boldsymbol{F}^T\boldsymbol{B}\right)\right)\right)\right)$. To this end, the computation of $\boldsymbol{m}$ in Eq. (4) is optimized as

$$\boldsymbol{m} = \beta\left(V\left(\Lambda^{-1}\left(V^T\left(\boldsymbol{F}^T\boldsymbol{B}\right)\right)\right)\right). \tag{6}$$

B. Details of Experiment Setup

In this section, we include more details of our experiment setup, including the source models and target datasets.

Implementation Details. Our implementation is based on MMDetection [1] with PyTorch 1.8 [8] and all experiments are conducted on 8 V100 GPUs. The base feature level l_0 in *Pyramid Feature Matching* is set as 3. The ground truth ranking of these detectors are obtained by fine-tuning all of them on the downstream tasks with well tuned training hyper-parameters. The overall Det-LogME algorithm is given in Algorithm 1.

Baseline Methods. We adopt 3 SOTA methods, KNAS [12], SFDA [11], and LogME [14], as the baseline methods and make comparisons with our proposed method. KNAS is a gradient based method different from recent efficient assessment method, we take it as a comparison with our gradient free approach. SFDA is the current SOTA method on

Algorithm 1 Det-LogME

Input: pre-trained detector $\mathcal{F}$, target dataset $\mathcal{D}_t$

Output: estimated transferability score Det-LogME

Extract multi-scale object-level features using pre-trained detector $\mathcal{F}$'s backbone followed by an ROIAlign layer and collect bounding box coordinates and class labels:
$\boldsymbol{F}\in\mathbb{R}^{M\times D}, \boldsymbol{B}\in\mathbb{R}^{M\times 4}, \boldsymbol{C}\in\mathbb{R}^{M}$

Find the match level features for all objects

Apply center normalization on $\boldsymbol{B}$ to obtain $\boldsymbol{B}^{cen}$

Unify $\boldsymbol{B}^{cen}$ and $\boldsymbol{C}$ as a unified label matrix $\boldsymbol{Y}^u$ by

$$\boldsymbol{Y}^u=\left[\left[\underbrace{(0,0,0,0)}_{1\text{ st}},\ldots,\underbrace{\overbrace{(x_c,y_c,w_c,h_c)}^{\boldsymbol{b}_i^{cen}}}_{c_i\text{ -th}},\ldots,\underbrace{(0,0,0,0)}_{K\text{ -th}}\right]\right]_M$$

Initialize $\alpha=1, \beta=1$, compute $\boldsymbol{F}^T\boldsymbol{F}=V\operatorname{diag}\{\sigma\}V^T$

while α and β not converge **do**

  Compute $\gamma=\sum_{i=1}^{D}\frac{\beta\sigma_i}{\alpha+\beta\sigma_i}, \Lambda=\operatorname{diag}\{(\alpha+\beta\sigma)\}$

  Compute $\boldsymbol{m}=\beta\left(V\left(\Lambda^{-1}\left(V^T\left(\boldsymbol{F}^T\boldsymbol{B}^{cen}\right)\right)\right)\right)$

  Expand $\boldsymbol{m}\in\mathbb{R}^D$ to $\boldsymbol{m}\in\mathbb{R}^{D\times(4\cdot K)}$ for matching $\boldsymbol{Y}^u$

  Update $\alpha\leftarrow\frac{\gamma}{\boldsymbol{m}^T\boldsymbol{m}}, \beta\leftarrow\frac{M-\gamma}{\|\boldsymbol{F}\boldsymbol{m}-\boldsymbol{B}^{cen}\|_2^2}$

end while

Compute U-LogME by

$$\begin{aligned}\text{U-LogME}=&\frac{M}{2}\log\beta+\frac{D}{2}\log\alpha-\frac{M}{2}\log 2\pi\\&-\frac{\beta}{2}\|\boldsymbol{F}\boldsymbol{m}-\boldsymbol{B}^{cen}\|_2^2-\frac{\alpha}{2}\boldsymbol{m}^T\boldsymbol{m}-\frac{1}{2}\log|A|,\end{aligned}$$

where $A=\alpha I+\beta\boldsymbol{F}^T\boldsymbol{F}$

Downsample $\boldsymbol{m}$ to $\boldsymbol{m}'\in\mathbb{R}^{D\times 4}$ by reserving the real coordinates of $\boldsymbol{B}^{cen}$, compute IoU-LogME $=\frac{|\boldsymbol{F}\boldsymbol{m}'\cap\boldsymbol{B}^{cen}|}{|\boldsymbol{F}\boldsymbol{m}'\cup\boldsymbol{B}^{cen}|}$

Compute Det-LogME $=$ U-LogME $+\mu\cdot$ IoU-LogME

Return Det-LogME

the classification task, so we formulate the multi-class object detection as a object-level classification task for adapting SFDA. LogME is the baseline of our work. Here, we describe the details for adapting these methods for object detection task.

KNAS is originally used for Neural Architecture Search (NAS) under a gradient kernel hypothesis. This hypothesis indicates that assuming $\boldsymbol{\mathcal{G}}$ is a set of all the gradients, there exists a gradient $\boldsymbol{g}$ which infers the downstream training performance. We adopt it as a gradient based approach to compare with our gradient free approach. Under this hypothesis, taking MSE loss for bounding box regression as an example, KNAS aims to minimize

$$\mathcal{L}(\boldsymbol{w})=\frac{1}{2}\left\|\hat{\boldsymbol{B}}-\boldsymbol{B}\right\|_2^2, \tag{7}$$

where $\boldsymbol{w}$ is the trainable weights, $\hat{\boldsymbol{B}}=[\hat{\boldsymbol{b}}_1,\ldots,\hat{\boldsymbol{b}}_M]^T$ is the bounding box prediction matrix, $\boldsymbol{B}=[\boldsymbol{b}_1,\ldots,\boldsymbol{b}_M]^T$ is the ground truth bounding box matrix, and M is the number of objects. Then gradient descent is applied to optimize the model weights:

$$\boldsymbol{\Theta}(t+1)=\boldsymbol{\Theta}(t)-\eta\frac{\partial\mathcal{L}(\boldsymbol{\Theta}(t))}{\partial\boldsymbol{\Theta}(t)}, \tag{8}$$

where t represents the t-th iteration and η is the learning rate. The gradient for an object sample i is

$$\frac{\partial\mathcal{L}(\boldsymbol{\Theta}(t),i)}{\partial(\boldsymbol{\Theta}(t))}=\left(\hat{\boldsymbol{b}}_i-\boldsymbol{b}_i\right)\frac{\partial\hat{\boldsymbol{b}}_i}{\partial\boldsymbol{\Theta}(t)}. \tag{9}$$

Then, a Gram matrix $\boldsymbol{H}$ is defined where the entry (i,j) is

$$\boldsymbol{H}_{i,j}(t)=\left(\frac{\partial\hat{\boldsymbol{b}}_j(t)}{\partial\boldsymbol{\Theta}(t)}\right)\left(\frac{\partial\hat{\boldsymbol{b}}_i(t)}{\partial\boldsymbol{\Theta}(t)}\right)^T. \tag{10}$$

$\boldsymbol{H}_{i,j}(t)$ is the dot-product between two gradient vectors $\boldsymbol{g}_i=\frac{\partial\hat{\boldsymbol{b}}_i(t)}{\partial\boldsymbol{\Theta}(t)}$ and $\boldsymbol{g}_j=\frac{\partial\hat{\boldsymbol{b}}_j(t)}{\partial\boldsymbol{\Theta}(t)}$. To this end, the gradient kernel $\boldsymbol{g}$ can be computed as the mean of all elements in the Gram matrix $\boldsymbol{H}$:

$$\boldsymbol{g}=\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial\hat{\boldsymbol{b}}_j(t)}{\partial\boldsymbol{\Theta}(t)}\right)\left(\frac{\partial\hat{\boldsymbol{b}}_i(t)}{\partial\boldsymbol{\Theta}(t)}\right)^T. \tag{11}$$

As the length of the whole gradient vector is too long, Eq. (11) is approximated by

$$\boldsymbol{g}=\frac{1}{QM^2}\sum_{q=1}^{Q}\sum_{i=1}^{M}\sum_{j=1}^{M}\left(\frac{\partial\hat{\boldsymbol{b}}_j(t)}{\partial\hat{\boldsymbol{\Theta}}^q(t)}\right)\left(\frac{\partial\hat{\boldsymbol{b}}_i(t)}{\partial\hat{\boldsymbol{\Theta}}^q(t)}\right)^T. \tag{12}$$

where Q is the number of layers in the detection head, and $\hat{\boldsymbol{\Theta}}^q$ is the sampled parameters from q-th layer and the length of $\hat{\boldsymbol{\Theta}}^q$ is set as 1000 in our implementation. The obtained gradient kernel $\boldsymbol{g}$ is regarded as the transferability score from KNAS.

SFDA is specially designed to assess the transferability for classification tasks, which is not applicable for single-class detection datasets used in this work including SKU-110K [3], WIDER FACE [13], and CrowdHuman [10]. It aims to leverage the neglected fine-tuning dynamics for transferability evaluation, which degrades the efficiency. Given object-level feature matrix $\boldsymbol{F}=[\boldsymbol{f}_1,\ldots,\boldsymbol{f}_M]^T$, with corresponding class label matrix $\boldsymbol{C}$, we consider object detection as an object-level multi-class classification task for adapting SFDA.

To utilize the fine-tuning dynamics, SFDA transforms the object feature matrix $\boldsymbol{F}$ to a space with good class separation under Regularized Fisher Discriminant Analysis (Reg-FDA). A transformation is defined to project $\boldsymbol{F}\in\mathbb{R}^{M\times D}$ to $\tilde{\boldsymbol{F}}\in\mathbb{R}^{M\times D'}$ by a projection matrix $\boldsymbol{U}\in\mathbb{R}^{D\times D'}$ with $\tilde{\boldsymbol{F}}:=\boldsymbol{U}^T\boldsymbol{F}$. The project matrix is

$$\boldsymbol{U}=\arg\max_{\boldsymbol{U}}\frac{d_b(\boldsymbol{U})}{d_w(\boldsymbol{U})}\overset{\text{def}}{=}\frac{\left|\boldsymbol{U}^\top\boldsymbol{S}_b\boldsymbol{U}\right|}{\left|\boldsymbol{U}^\top\left[(1-\lambda)\boldsymbol{S}_w+\lambda\boldsymbol{I}\right]\boldsymbol{U}\right|}, \tag{13}$$

Table 1. Ranking results of of six methods for 1% 33-choose-22 possible source model sets (over 1.9M) on 6 downstream target datasets. Higher ρ_w and Recall@1 indicate better ranking and transferability metric. As SFDA is specifically designed for classification task, it is not applicable for the single-class task of CrowdHuman. The results of all three variants of our approach, U-LogME, IoU-LogME, and Det-LogME are reported. The best methods are in red and good ones are in blue.

Measure	Weighted Pearson's Coefficient (ρ_w)						Recall@1					
Method	KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME
Pascal VOC	0.01±0.15	**0.71±0.14**	-0.04±0.16	-0.07±0.23	**0.73±0.13**	0.68±0.12	0.26±0.44	0.33±0.47	**0.53±0.50**	0.20±0.40	0.34±0.47	**0.41±0.49**
CityScapes	0.15±0.18	0.46±0.11	0.38±0.09	0.19±0.13	**0.53±0.10**	**0.55±0.09**	**0.53±0.50**	0.00±0.00	**0.53±0.50**	0.12±0.33	**0.53±0.50**	**0.53±0.50**
SODA	-0.11±0.21	0.60±0.13	0.28±0.13	0.12±0.13	**0.65±0.12**	**0.66±0.11**	0.00±0.00	0.00±0.00	**0.53±0.50**	0.12±0.33	**0.53±0.50**	**0.53±0.50**
CrowdHuman	-0.21±0.13	N/A	0.08±0.19	0.11±0.17	**0.31±0.08**	**0.30±0.08**	0.00±0.00	N/A	**0.65±0.48**	0.58±0.49	**0.65±0.48**	**0.65±0.48**
VisDrone	0.15±0.21	0.29±0.15	0.35±0.10	0.12±0.10	**0.44±0.12**	**0.44±0.11**	0.12±0.32	0.34±0.47	0.17±0.38	0.01±0.11	**0.25±0.43**	**0.25±0.43**
DeepLesion	0.08±0.18	-0.37±0.29	0.34±0.20	**0.54±0.19**	-0.17±0.34	**0.50±0.16**	0.01±0.09	0.00±0.00	0.26±0.44	**0.57±0.50**	0.00±0.03	**0.42±0.49**
Average	0.01±0.18	0.34±0.16	0.23±0.15	0.20±0.16	**0.42±0.15**	**0.52±0.11**	0.15±0.36	0.11±0.31	**0.44±0.50**	0.27±0.44	0.38±0.49	**0.46±0.50**

Table 2. The transferability scores obtained from 6 metrics and fine-tuning mAP on Pascal VOC and CityScapes datasets. The last row is the corresponding ranking correlation τ_w for every metric.

Model	Backbone	Pascal VOC							CityScapes						
		KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP	KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP
Faster RCNN	R50	2.326E-01	0.791	-6.193	-3.223	0.482	1.199	84.5	-2.093E+00	0.879	-6.257	-1.518	0.624	9.229	41.9
	R101	1.095E-01	0.809	-6.177	-3.160	0.492	1.258	84.5	-1.791E+00	0.887	-6.258	-1.478	0.624	9.289	42.3
	X101-32x4d	-4.396E-02	0.822	-6.146	-2.969	0.505	1.380	85.2	-2.386E+00	0.892	-6.269	-1.397	0.622	9.242	43.5
	X101-64x4d	9.018E-01	0.825	-6.129	-2.944	0.509	1.405	85.6	-1.664E+00	0.894	-6.270	-1.381	0.624	9.353	42.8
Cascade RCNN	R50	-6.438E-01	0.795	-6.232	-3.203	0.481	1.206	84.1	-6.218E+00	0.874	-6.286	-1.514	0.618	9.013	44.1
	R101	-2.226E-01	0.811	-6.222	-3.176	0.490	1.247	84.9	-6.553E+00	0.883	-6.289	-1.489	0.621	9.127	43.7
	X101-32x4d	-6.405E-01	0.826	-6.194	-3.024	0.503	1.351	85.6	-5.763E+00	0.891	-6.297	-1.415	0.620	9.160	44.1
	X101-64x4d	1.270E+00	0.831	-6.190	-3.006	0.505	1.367	85.8	-5.182E+00	0.891	-6.290	-1.402	0.620	9.166	45.4
Dynamic RCNN	R50	-8.148E-03	0.791	-6.206	-2.875	0.483	1.343	84.0	1.878E-01	0.869	-6.303	-1.352	0.617	9.110	42.5
RegNet	400MF	5.056E-01	0.750	-6.162	-3.387	0.465	1.076	83.3	2.401E-01	0.845	-6.264	-1.647	0.606	8.400	39.9
	800MF	6.691E-02	0.758	-6.156	-3.295	0.468	1.122	83.9	-2.308E+00	0.855	-6.279	-1.606	0.606	8.441	40.3
	1.6GF	1.523E-01	0.770	-6.162	-3.232	0.472	1.161	84.6	-1.504E+00	0.869	-6.279	-1.553	0.613	8.767	41.8
	3.2GF	6.241E-02	0.786	-6.170	-3.186	0.482	1.215	85.5	-3.148E-01	0.877	-6.269	-1.527	0.618	8.984	42.7
	4GF	3.995E-01	0.790	-6.166	-3.133	0.484	1.242	85.0	-1.451E+00	0.878	-6.278	-1.506	0.617	8.956	43.1
DCN	R50	3.852E-02	0.825	-6.122	-2.748	0.511	1.490	86.1	-6.647E-01	0.889	-6.246	-1.267	0.625	9.497	42.6
	R101	-9.254E-02	0.836	-6.155	-2.812	0.516	1.481	86.5	-2.072E+00	0.894	-6.253	-1.298	0.626	9.503	43.1
	X101-32x4d	7.048E-02	0.846	-6.100	-2.653	0.525	1.577	86.9	-7.308E-01	0.899	-6.253	-1.227	0.626	9.571	43.5
FCOS	R50	1.023E+01	0.289	-6.093	-1.856	0.264	0.988	77.3	6.343E+00	0.492	-6.434	-0.992	0.491	4.318	40.4
	R101	5.233E+00	0.280	-6.032	-2.101	0.262	0.884	79.4	5.277E+00	0.515	-6.426	-1.124	0.491	4.219	41.2
RetinaNet	R18	-3.404E-01	0.733	-6.289	-2.928	0.442	1.177	80.9	1.157E-02	0.844	-6.411	-1.438	0.597	8.206	36.7
	R50	-1.357E-01	0.759	-6.277	-2.975	0.457	1.213	84.1	5.439E-03	0.867	-6.370	-1.388	0.606	8.609	40.0
	R101	-1.807E-01	0.774	-6.259	-2.972	0.467	1.246	84.4	4.709E-02	0.879	-6.357	-1.374	0.612	8.854	40.6
	X101-32x4d	-1.030E-01	0.792	-6.260	-2.763	0.475	1.360	84.6	2.935E-02	0.881	-6.377	-1.308	0.608	8.762	41.2
	X101-64x4d	-3.170E-01	0.792	-6.229	-2.722	0.475	1.376	85.3	2.304E-02	0.886	-6.366	-1.276	0.610	8.858	42.0
Sparse RCNN	R50	9.846E+03	0.777	-6.267	-3.243	0.456	1.102	84.7	-1.640E+04	0.878	-6.414	-1.595	0.602	8.304	38.9
	R101	-2.104E+04	0.795	-6.238	-3.263	0.466	1.127	85.0	1.198E+04	0.884	-6.396	-1.601	0.602	8.304	39.3
Deformable DETR	R50	8.873E+03	0.794	-5.221	-2.295	0.462	1.501	87.0	1.363E+05	0.881	-5.376	-1.065	0.673	11.602	45.5
Faster RCNN OI	R50	2.038E+00	0.724	-6.016	-4.100	0.443	0.716	82.2	3.288E+00	0.837	-6.260	-1.951	0.602	7.982	39.3
RetinaNet OI	R50	-2.045E-01	0.697	-6.195	-3.335	0.430	0.974	82.0	1.701E-01	0.845	-6.343	-1.624	0.600	8.177	39.5
SoCo	R50	-3.222E+00	0.703	-6.094	-3.062	0.433	1.093	56.5	4.629E+01	0.836	-6.237	-1.473	0.606	8.536	41.7
InsLoc	R50	-3.153E-03	0.566	-6.239	-1.592	0.424	1.649	86.7	1.041E-02	0.756	-6.322	-0.738	0.582	8.191	40.3
UP-DETR	R50	8.225E+02	0.175	-6.267	-3.086	0.238	0.404	59.3	-2.994E+02	0.399	-6.485	-1.404	0.403	0.455	30.9
DETReg	R50	-6.832E+02	0.189	-5.999	-3.872	0.248	0.129	63.5	-9.335E+02	0.427	-5.892	-1.958	0.440	1.462	38.7
τ_w		0.15	0.64	0.22	0.43	0.54	**0.79**	N/A	-0.02	0.51	0.32	0.18	0.68	**0.71**	N/A

where $d_b(\boldsymbol{U})$ and $d_w(\boldsymbol{U})$ represent between scatter of classes and within scatter of each class, $\lambda \in [0, 1]$ is a regularization coefficient for a trade-off between the inter-class separation and intra-class compactness, and $\boldsymbol{I}$ is an identity matrix. The between and within scatter matrix $\boldsymbol{S}_b$ and $\boldsymbol{S}_w$ are difined as

$$\begin{aligned} \boldsymbol{S}_b &= \sum_{c=1}^{K} M_c \left(\nu_c - \nu\right)\left(\nu_c - \nu\right)^\top \\ \boldsymbol{S}_w &= \sum_{c=1}^{K}\sum_{i=1}^{M_c} \left(\boldsymbol{f}_i^{(c)} - \nu_c\right)\left(\boldsymbol{f}_i^{(c)} - \nu_c\right)^\top, \end{aligned} \tag{14}$$

where $\nu = \sum_{i=1}^{M} \boldsymbol{f}_i$ and $\nu_c = \sum_{i=1}^{M} \boldsymbol{f}_i^{(c)}$ are the mean of all and c-th class object features.

With the intuition that a model with Infomin requires stronger supervision for minimizing within scatter of every class which results in better classes separation. λ is instantiated by $\lambda = \exp^{-a\sigma(\boldsymbol{S}_w)}$, where a is a positive constant and $\sigma(\boldsymbol{S}_w)$ is the largest eigenvalue of $\boldsymbol{S}_w$. For every class, SFDA assumes $\tilde{\boldsymbol{f}}_i^{(c)} \sim \mathcal{N}\left(\boldsymbol{U}^\top \nu_c, \Sigma_c\right)$, where Σ_c is the covariance matrix of $\{\tilde{\boldsymbol{f}}_i^{(c)}\}_{i=1}^{M_c}$. With projection matrix $\boldsymbol{U}$, the score function for class c is

$$\delta_c\left(\boldsymbol{f}_i\right) = \boldsymbol{f}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \nu_c - \frac{1}{2}\nu_c^\top \boldsymbol{U}\boldsymbol{U}^\top \nu_c + \log\frac{M_c}{M}. \tag{15}$$

Then, the final class prediction probability is obtained by

Table 3. The transferability scores obtained from 6 metrics and fine-tuning mAP on SODA and CrowdHuman datasets. The last row is the corresponding ranking correlation τ_w for every metric.

Model	Backbone	SODA							CrowdHuman						
		KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP	KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP
Faster RCNN	R50	-1.314E+00	0.831	-5.698	-2.148	0.542	16.546	34.7	-1.216E+01	N/A	-6.660	-0.116	0.575	1.357	41.4
	R101	-2.636E+00	0.846	-5.679	-2.071	0.548	17.121	35.0	-1.648E+01	N/A	-6.655	-0.111	0.577	1.482	41.3
	X101-32x4d	-2.516E+00	0.852	-5.664	-1.924	0.554	17.643	35.7	-7.494E+00	N/A	-6.620	-0.074	0.586	2.081	41.2
	X101-64x4d	-1.226E+00	0.856	-5.660	-1.914	0.557	17.900	36.4	-9.416E+00	N/A	-6.596	-0.045	0.592	2.460	41.5
Cascade RCNN	R50	-9.109E+00	0.826	-5.746	-2.129	0.537	16.183	35.3	-1.958E+01	N/A	-6.681	-0.138	0.568	0.851	43.0
	R101	-1.177E+01	0.836	-5.733	-2.091	0.543	16.685	35.9	-2.723E+01	N/A	-6.683	-0.143	0.567	0.781	42.8
	X101-32x4d	-1.509E+01	0.846	-5.709	-1.966	0.549	17.245	36.8	-1.755E+01	N/A	-6.662	-0.126	0.573	1.170	43.2
	X101-64x4d	-1.277E+01	0.851	-5.733	-1.946	0.552	17.453	37.4	-1.083E+01	N/A	-6.660	-0.121	0.573	1.225	43.7
Dynamic RCNN	R50	-1.597E+00	0.820	-5.758	-1.871	0.535	16.169	35.2	-5.448E+00	N/A	-6.674	-0.130	0.568	0.900	41.8
RegNet	400MF	-1.900E+00	0.785	-5.670	-2.300	0.520	14.767	32.5	-1.393E+01	N/A	-6.628	-0.099	0.579	1.619	38.0
	800MF	-1.512E+00	0.803	-5.694	-2.239	0.525	15.206	34.2	-8.908E+00	N/A	-6.636	-0.099	0.578	1.546	39.8
	1.6GF	-2.576E+00	0.815	-5.668	-2.169	0.537	16.190	35.7	-1.558E+01	N/A	-6.653	-0.118	0.573	1.215	41.8
	3.2GF	-2.007E+00	0.827	-5.682	-2.140	0.538	16.288	37.0	-1.560E+01	N/A	-6.647	-0.106	0.577	1.438	41.7
	4GF	-1.735E+00	0.826	-5.700	-2.097	0.539	16.380	37.0	-1.600E+01	N/A	-6.650	-0.111	0.576	1.388	41.9
DCN	R50	-1.077E+00	0.844	-5.677	-1.736	0.553	17.632	35.3	-1.562E+01	N/A	-6.569	-0.013	0.602	3.121	43.1
	R101	-8.408E-01	0.846	-5.669	-1.786	0.556	17.874	35.3	-1.073E+01	N/A	-6.584	-0.033	0.596	2.718	43.4
	X101-32x4d	-1.797E+00	0.859	-5.676	-1.655	0.559	18.189	36.0	-7.181E+00	N/A	-6.494	0.018	0.614	3.876	44.3
FCOS	R50	-1.287E-02	0.510	-5.729	-1.328	0.415	6.902	33.3	-1.263E+00	N/A	-6.578	-0.040	0.570	1.046	35.6
	R101	6.980E-01	0.541	-5.688	-1.535	0.413	6.610	34.5	-1.601E+00	N/A	-6.537	-0.006	0.578	1.593	36.8
RetinaNet	R18	-2.071E-02	0.778	-5.846	-1.988	0.510	14.099	29.6	3.133E-02	N/A	-6.696	-0.161	0.554	0.008	35.8
	R50	-7.809E-01	0.818	-5.817	-1.885	0.527	15.528	33.9	3.080E-02	N/A	-6.702	-0.168	0.555	0.035	38.3
	R101	-1.857E-02	0.828	-5.835	-1.874	0.532	15.877	34.0	-1.330E-03	N/A	-6.699	-0.164	0.556	0.121	38.6
	X101-32x4d	-1.033E-01	0.833	-5.864	-1.794	0.530	15.766	34.2	-4.154E-03	N/A	-6.691	-0.157	0.557	0.171	38.9
	X101-64x4d	-2.995E-01	0.839	-5.851	-1.717	0.537	16.430	35.6	8.897E-03	N/A	-6.676	-0.147	0.562	0.477	39.9
Sparse RCNN	R50	-5.855E+04	0.824	-5.892	-2.256	0.518	14.636	35.9	4.174E+04	N/A	-6.683	-0.154	0.554	0.009	38.6
	R101	-1.934E+05	0.833	-5.869	-2.255	0.523	14.991	36.3	-1.077E+03	N/A	-6.676	-0.145	0.557	0.190	39.2
Deformable DETR	R50	-9.691E+04	0.820	-4.557	-1.421	0.589	20.697	38.8	-3.523E+04	N/A	-5.447	0.904	0.790	15.536	45.3
Faster RCNN OI	R50	-2.170E+01	0.767	-5.543	-2.756	0.513	13.942	32.8	-6.768E+01	N/A	-6.533	-0.006	0.601	3.037	40.0
RetinaNet OI	R50	-2.793E-01	0.780	-5.782	-2.278	0.507	13.741	33.4	7.107E-03	N/A	-6.650	-0.104	0.571	1.077	38.5
SoCo	R50	-5.681E-01	0.759	-5.584	-2.074	0.517	14.607	33.2	-1.681E+01	N/A	-6.553	0.019	0.601	3.065	40.6
InsLoc	R50	1.857E-02	0.691	-5.750	-0.842	0.490	13.092	31.4	7.323E-01	N/A	-6.652	-0.117	0.565	0.690	40.7
UP-DETR	R50	-4.658E+02	0.457	-6.040	-1.946	0.338	0.423	20.1	3.661E+02	N/A	-6.613	-0.145	0.555	0.062	35.4
DETReg	R50	-8.563E+02	0.467	-5.584	-2.477	0.371	2.783	24.3	-6.686E+02	N/A	-6.202	0.221	0.638	5.498	41.0
τ_w		-0.44	0.43	0.22	0.03	**0.66**	0.65	N/A	-0.47	N/A	0.37	0.39	**0.51**	**0.51**	N/A

normalizing $\{\delta_c(\boldsymbol{f}_i)\}_K$ with softmax function:

$$p(c_i \mid \boldsymbol{f}_i) = \frac{\exp^{\delta_{c_i}(\boldsymbol{f}_i)}}{\sum_{c=1}^{K} \exp^{\delta_c(\boldsymbol{f}_i)}} \tag{16}$$

To this end, the transferability score is expressed as the mean of $p(c_i \mid \boldsymbol{f}_i)$ over all object samples by

$$p(\boldsymbol{C} \mid \boldsymbol{F}) = \frac{1}{M} \sum_{i=1}^{M} \frac{\exp^{\delta_{c_i}(\boldsymbol{f}_i)}}{\sum_{c=1}^{K} \exp^{\delta_c(\boldsymbol{f}_i)}}. \tag{17}$$

LogME is following Eq. (3) described in Sec. A.

C. More Experimental Results

Ranking Performance. Except for Weighted Kendall's tau (τ_w) and Top-1 Relative Accuracy (Rel@1), we also evaluate the transferability metrics based on Weighted Pearson's coefficient (ρ_w) [2] and Recall@1 [7], as shown in Table 1. Weighted Pearson's coefficient is used to measure the linear correlation between transferability scores and ground truth fine-tuning performance. Recall@1 is used to measure the ratio of successfully selecting the model with best fine-tuning performance. The evaluation is conducted on 1% 33-choose-22 possible source model sets (over 1.9M). Regarding ρ_w, we can draw the conclusion that Det-LogME outperforms all three SOTA methods consistently on 6 downstream tasks by a large margin. The IoU based metric IoU-LogME also performs well on 5 datasets. Regarding Recall@1, our proposed Det-LogME outperforms previous SOTA methods in average.

Detailed Ranking Results. We provide detailed raw ranking results of all 33 pre-trained detectors on 6 downstream tasks, including the transferability scores, ground truth performance (the average result of 3 runs with very light variance), and *Weighted Kendall*'s tau τ_w. The results are provided in the following tables. Table 2 shows results on Pascal VOC and CityScapes, Table 3 shows results on SODA and CrowdHuman, and Table 4 contains results on VisDrone and DeepLesion.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[2] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007. 4

[3] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5236, 2019. 2

Table 4. The transferability scores obtained from 6 metrics and fine-tuning mAP on VisDrone and DeepLesion datasets. The last row is the corresponding ranking correlation τ_w for every metric.

Model	Backbone	VisDrone							DeepLesion						
		KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP	KNAS	SFDA	LogME	U-LogME	IoU-LogME	Det-LogME	mAP
Faster RCNN	R50	3.634E-01	0.645	-6.611	-1.771	0.432	1.357	21.3	6.270E-01	0.665	-4.858	-3.203	0.394	1.114	2.9
	R101	-3.928E-01	0.662	-6.608	-1.734	0.438	1.482	21.5	6.524E-01	0.675	-4.801	-3.118	0.396	1.129	2.8
	X101-32x4d	-1.176E+00	0.661	-6.551	-1.648	0.442	2.081	22.3	4.738E-01	0.691	-4.774	-2.838	0.392	1.137	3.0
	X101-64x4d	-5.828E-01	0.669	-6.527	-1.636	0.443	2.460	23.2	3.634E-01	0.682	-4.778	-2.967	0.386	1.107	3.7
Cascade RCNN	R50	-8.062E-01	0.637	-6.652	-1.775	0.427	0.851	20.7	-1.159E+00	0.662	-4.795	-3.041	0.387	1.102	3.1
	R101	-2.153E+00	0.645	-6.649	-1.764	0.428	0.781	21.4	-1.697E+00	0.674	-4.752	-3.295	0.392	1.100	3.1
	X101-32x4d	-2.963E+00	0.659	-6.607	-1.687	0.436	1.170	21.9	-7.776E-01	0.685	-4.717	-2.981	0.395	1.133	3.6
	X101-64x4d	-4.002E+00	0.662	-6.600	-1.670	0.438	1.225	22.5	-1.154E+00	0.678	-4.667	-3.202	0.385	1.083	3.0
Dynamic RCNN	R50	1.827E-01	0.639	-6.629	-1.650	0.433	0.900	16.1	9.446E-01	0.652	-4.712	-1.752	0.396	1.237	3.0
RegNet	400MF	-9.387E-01	0.609	-6.497	-1.902	0.433	1.619	19.2	7.476E-01	0.660	-4.826	-2.895	0.392	1.131	2.8
	800MF	-6.771E-01	0.631	-6.552	-1.860	0.437	1.546	21.1	1.511E-01	0.641	-4.871	-2.528	0.392	1.162	2.9
	1.6GF	-2.191E-01	0.646	-6.588	-1.831	0.438	1.215	22.2	5.915E-01	0.666	-4.770	-2.723	0.403	1.183	3.1
	3.2GF	-1.319E+00	0.657	-6.584	-1.801	0.442	1.438	23.3	4.484E-01	0.658	-4.790	-2.486	0.397	1.182	3.3
	4GF	-2.118E+00	0.654	-6.572	-1.785	0.442	1.388	23.2	1.133E+00	0.642	-4.833	-2.539	0.396	1.173	2.8
DCN	R50	-1.394E+00	0.654	-6.513	-1.582	0.447	3.121	21.7	3.988E-01	0.705	-4.570	-2.418	0.425	1.282	2.7
	R101	-1.431E+00	0.666	-6.529	-1.623	0.447	2.718	21.9	-7.920E-01	0.707	-4.602	-2.527	0.431	1.293	3.0
	X101-32x4d	-3.897E-01	0.677	-6.397	-1.537	0.458	3.876	23.3	3.103E-01	0.698	-4.573	-2.023	0.421	1.300	3.5
FCOS	R50	5.389E+00	0.476	-6.523	-1.362	0.393	1.046	21.6	-5.430E+00	0.254	-4.747	6.207	0.295	1.542	4.5
	R101	5.258E+00	0.493	-6.448	-1.474	0.396	1.593	22.4	-6.504E+00	0.202	-4.352	5.016	0.283	1.405	4.8
RetinaNet	R18	-4.476E-02	0.603	-6.687	-1.712	0.419	0.008	14.7	-4.071E-02	0.525	-4.836	-1.504	0.410	1.306	2.8
	R50	-1.586E-02	0.645	-6.695	-1.644	0.427	0.035	17.9	-1.368E-02	0.513	-4.871	-1.081	0.389	1.268	3.4
	R101	-8.479E-02	0.650	-6.678	-1.637	0.430	0.121	18.2	1.456E-01	0.569	-4.822	-1.087	0.416	1.360	3.7
	X101-32x4d	-2.029E-01	0.647	-6.681	-1.600	0.427	0.171	18.5	6.097E-01	0.515	-4.824	-0.987	0.412	1.355	4.5
	X101-64x4d	-9.568E-02	0.662	-6.645	-1.538	0.434	0.477	19.1	1.775E-01	0.541	-4.857	-0.628	0.412	1.383	4.2
Sparse RCNN	R50	-2.382E+03	0.643	-6.695	-1.887	0.425	0.009	14.3	1.085E+04	0.652	-4.905	-1.803	0.402	1.252	3.7
	R101	4.137E+02	0.653	-6.683	-1.891	0.429	0.190	14.2	1.153E+04	0.613	-4.889	-2.000	0.392	1.204	3.8
Deformable DETR	R50	4.226E+05	0.614	-5.226	-1.435	0.476	15.536	23.3	8.210E+04	0.660	-4.327	-2.108	0.425	1.307	2.8
Faster RCNN OI	R50	-2.383E+00	0.614	-6.401	-2.036	0.434	3.037	20.2	1.880E+00	0.656	-4.772	-5.944	0.373	0.820	2.5
RetinaNet OI	R50	-8.290E-01	0.630	-6.627	-1.792	0.425	1.077	16.3	-3.031E-01	0.655	-4.784	-2.507	0.375	1.105	3.1
SoCo	R50	-1.907E+01	0.612	-6.568	-1.656	0.427	3.065	20.6	-2.707E+00	0.676	-4.781	-3.650	0.391	1.068	2.3
InsLoc	R50	-7.507E-02	0.540	-6.625	-0.925	0.417	0.690	18.8	-1.149E-02	0.496	-4.895	1.239	0.388	1.452	0.5
UP-DETR	R50	-4.091E+02	0.428	-6.851	-1.341	0.345	0.062	13.5	-5.645E+03	0.217	-5.985	-4.089	0.134	0.167	0.4
DETReg	R50	-1.594E+02	0.419	-5.978	-1.963	0.367	5.498	15.1	-8.055E+03	0.411	-5.119	-4.957	0.272	0.562	2.0
τ_w		0.16	0.53	0.52	0.14	**0.71**	**0.71**	N/A	-0.14	-0.30	0.13	**0.61**	-0.09	0.50	N/A

[4] Stephen F. Gull. *Developments in Maximum Entropy Data Analysis*, pages 53–71. Springer Netherlands, Dordrecht, 1989. 1

[5] Kevin H Knuth, Michael Habeck, Nabin K Malakar, Asim M Mubeen, and Ben Placek. Bayesian evidence and model selection. *Digital Signal Processing*, 47:50–67, 2015. 1

[6] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 1

[7] Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2663–2673, 2021. 4

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[10] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018. 2

[11] Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. *European Conference on Computer Vision*, 2022. 1

[12] Jingjing Xu, Liang Zhao, Junyang Lin, Rundong Gao, Xu Sun, and Hongxia Yang. Knas: green neural architecture search. In *International Conference on Machine Learning*, pages 11613–11625. PMLR, 2021. 1

[13] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5525–5533, 2016. 2

[14] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR, 2021. 1