A. Derivation for Mutual Information Framework

This section describes the detailed derivation for our mutual information framework. For clarity, we list the notations in Tab. 6.

A.1. Single-input Sinle-target Pre-training

We start with the basic form of single-input single-target pre-training. The desired objective is to maximize the conditional mutual information between the input representation z_x and the target representation z_y given the input transform t_x and target transform t_y:

$$\max I(z_x; z_y \mid t_x, t_y). \tag{5}$$

According to the definition of conditional mutual information, we have

$$\begin{aligned}
&I(z_x; z_y \mid t_x, t_y) \\
&= \int p(t_x, t_y) \int \Big[p(z_x, z_y \mid t_x, t_y) \cdot \\
&\qquad \log \frac{p(z_y \mid z_x, t_x, t_y)}{p(z_y \mid t_x, t_y)} \Big] dz_x dz_y dt_x dt_y \\
&= \int p(t_x, t_y) \int \Big[p(z_x, z_y \mid s, t_x, t_y) p(s \mid t_x, t_y) \cdot \\
&\qquad \log \frac{p(z_y \mid z_x, t_x, t_y)}{p(z_y \mid t_x, t_y)} \Big] dz_x dz_y dt_x dt_y ds \\
&= \int p(s, t_x, t_y) \int \Big[p(z_x \mid x) p(z_y \mid y) \cdot \\
&\qquad \log \frac{p(z_y \mid z_x, t_x, t_y)}{p(z_y \mid t_x, t_y)} \Big] dz_x dz_y dt_x dt_y ds \\
&= \mathbb{E}_{p(s, t_x, t_y, z_x)} \Big[\int p(z_y \mid y) \log p(z_y \mid z_x, t_x, t_y) dz_y \Big] \\
&\quad - \mathbb{E}_{p(s, t_x, t_y, z_x, z_y)} \Big[\log p(z_y \mid t_x, t_y) \Big] \\
&= -\underbrace{\mathbb{E}_{p(s, t_x, t_y, z_x)} \Big[H\Big(p(z_y \mid y), p(z_y \mid z_x, t_x, t_y) \Big) \Big]}_{\text{prediction term for target representation}} \\
&\quad + \underbrace{\mathbb{E}_{p(t_y)} \Big[H\Big(p(z_y \mid t_y) \Big) \Big]}_{\text{regularization term to avoid collapse}},
\end{aligned} \tag{6}$$

where the third equation holds because two representations are independent given the input and target, and in the last equation we apply the definitions of entropy and cross-entropy. Eq. (6) shows that the mutual information can be divided into a prediction term and a regularization term. The prediction term requires the predicted distribution to be close to the target distribution, while the regularization term requires the target representations to maintain high entropy.

Next, we introduce parameterization to actually compute these terms. Two representations are encoded via an input encoder f_θ and a target encoder f_ϕ, respectively. Because we do not know $p(z_y \mid z_x, t_x, t_y)$ in advance, we adopt an approximation by first predicting $\hat{z}_y = f_\psi(z_x, t_x, t_y)$ and then estimating with the posterior distribution $\hat{P}(z_y \mid \hat{z}_y)$. The mutual information thus becomes

$$\begin{aligned}
&I(z_x; z_y \mid t_x, t_y) \\
&= \int p(t_x, t_y) \int \Big[p(z_x \mid t_x, t_y) p(z_y \mid z_x, t_x, t_y) \cdot \\
&\qquad \log \frac{p(z_y \mid z_x, t_x, t_y)}{p(z_y \mid t_x, t_y)} \Big] dz_x dz_y dt_x dt_y \\
&= \mathbb{E}_{p(z_x, t_x, t_y)} \Big[\int p(z_y \mid z_x, t_x, t_y) \log p(z_y \mid z_x, t_x, t_y) dz_y \Big] \\
&\quad - \mathbb{E}_{p(z_x, z_y, t_x, t_y)} \Big[\log p(z_y \mid t_x, t_y) \Big] \\
&= \underbrace{\mathbb{E}_{p(z_x, t_x, t_y)} \Big[\int p(z_y \mid z_x, t_x, t_y) \log \frac{p(z_y \mid z_x, t_x, t_y)}{\hat{P}(z_y \mid \hat{z}_y)} dz_y \Big]}_{\text{KL Divergence} \geq 0} \\
&\quad + \mathbb{E}_{p(z_x, t_x, t_y)} \Big[\int p(z_y \mid z_x, t_x, t_y) \log \hat{P}(z_y \mid \hat{z}_y) dz_y \Big] \\
&\quad - \mathbb{E}_{p(z_x, z_y, t_x, t_y)} \Big[\log p(z_y \mid t_x, t_y) \Big] \\
&\geq \mathbb{E}_{p(z_x, t_x, t_y)} \Big[\int p(z_y \mid z_x, t_x, t_y) \log \hat{P}(z_y \mid \hat{z}_y) dz_y \Big] \\
&\quad - \mathbb{E}_{p(z_x, z_y, t_x, t_y)} \Big[\log p(z_y \mid t_x, t_y) \Big] \\
&= \mathbb{E}_{p(z_x, z_y, t_x, t_y)} \Big[\log \hat{P}(z_y \mid \hat{z}_y) \cdot \underbrace{\int p(s \mid z_x, z_y, t_x, t_y) ds}_{\text{the integral is equal to 1}} \Big] \\
&\quad - \mathbb{E}_{p(t_y)} \Big[\int p(z_y \mid t_y) \log p(z_y \mid t_y) \cdot \\
&\qquad \underbrace{\int p(z_x, t_x \mid z_y, t_y) dz_x dt_x}_{\text{this integral is equal to 1}} dz_y \Big] \\
&= \underbrace{\mathbb{E}_{p(s, t_x, t_y)} \Big[\log \hat{P}\Big(z_y(\phi) \mid \hat{z}_y(\theta, \psi) \Big) \Big]}_{\text{prediction term for target representation}} \\
&\quad + \underbrace{\mathbb{E}_{p(t_y)} \Big[H(z_y(\phi) \mid t_y) \Big]}_{\text{regularization term to avoid collapse}},
\end{aligned} \tag{7}$$

where the fourth inequality holds because KL Divergence will not be less than 0. In the fifth equality, we introduce training sample s to the expectation of the first term and move z_x and t_x from the expectation of the second term. In the last equality, z_x and z_y is moved out of the

Pre-training Method	Typical Work	Input Data x	Target Data y	Input Representation z_x	Target Representation z_y	Regularization $H\big(p(z_y\|t_y)\big)$	Distribution Form $\hat{P}$
Supervised Pre-training :							
Image Classification	ViT [24]	view1	category	dense feature	category embedding	negative categories	Boltzmann
Weakly-supervised Pre-training :							
Contrastive Language-Image Pre-training	CLIP [55]	view1	text	dense feature	text embedding	negative texts	Boltzmann
Self-supervised Pre-training (intra-view) :							
Auto-Encoder	-	view1	view1	dense feature	dense pixels	-	Gaussian
[1]Dense Distillation	FD [81],BEiT v2 tokenizer [54]	view1	view1	dense feature	dense feature	stop gradient	Gaussian
Global Distillation	-	view1	view1	dense feature	global feature	stop gradient	Boltzmann
Masked Image Modeling$_{\text{pixel}}$	MAE [30]	masked view1	view1	dense feature	dense pixels	-	Gaussian
[2]Masked Image Modeling$_{\text{feature}}$	data2vec [4],MILAN [35], BEiT [5],BEiT v2 [54]	masked view1	view1	dense feature	dense feature	stop gradient	Gaussian
Masked Image Modeling$_{\text{global}}$	-	masked view1	view1	dense feature	global feature	stop gradient	Gaussian
Self-supervised Pre-training (inter-view) :							
Novel View Synthesis	-	view2	view1	dense feature	dense pixels	-	Gaussian
Dense Instance Discrimination	DenseCL [80]	view2	view1	dense feature	dense feature	negative samples	Boltzmann
[3]Instance Discrimination	MoCo [31],BYOL [27], Barlow Twins [91]	view 2	view1	dense feature	global feature	negative samples / stop gradient / decorrelation	Boltzmann / Gaussian
Siamese Image Modeling$_{\text{pixel}}$	-	masked view2	view1	dense feature	dense pixels	-	Gaussian
Siamese Image Modeling$_{\text{feature}}$	SiameseIM [67]	masked view2	view1	dense feature	dense feature	stop gradient	Gaussian
Siamese Image Modeling$_{\text{global}}$	MSN [3]	masked view2	view1	dense feature	global feature	negative samples	Boltzmann

Table 5. Instances of our mutual information based pre-training framework. We only include single-input single-target methods in this table. Methods without a listed typical work have rarely been explored before in pre-training. [1]Input representation of Dense Distillation can be continuous (FD) or discrete (BEiT v2 tokenizer). [2]Target encoder of Masked Image Modeling$_{\text{feature}}$ can be momentum encoder (data2vec), pre-trained image encoder (MILAN), dVAE (BEiT), or discrete tokenizer distilled from pre-trained encoders (BEiT v2). [3]Regularization term of Instance Discrimination can be negative samples (MoCo), stop-gradient (BYOL), or decorrelation (Barlow Twins).

Notation	Meaning	Typical Choices in Vision-centric Pre-training Paradigms		
		Supervised	Weakly-supervised	Self-supervised
s	**training sample** from the training dataset	image-category pair	image-text pair	image only
t_x	**input transform operation** applied to the sample s	apply image augmentation	apply image augmentation	apply image augmentation
t_y	**target transform operation** applied to the sample s	get annotated category	get paired text	apply image augmentation
$\boldsymbol{t}_x$	**the set of input transform operations** applied to the sample s	-	-	-
$\boldsymbol{t}_y$	**the set of target transform operations** applied to the sample s	-	-	-
$x = t_x(s)$	**input data** for the network training	augmented image	augmented image	augmented image
$y = t_y(s)$	**target data** for the network training	annotated category	paired text	augmented image
$\{x_i\}_{i=1}^{N} = \boldsymbol{t}_x(s)$	**multiple inputs** for the network training	-	-	-
$\{y_j\}_{j=1}^{M} = \boldsymbol{t}_y(s)$	**multiple targets** for the network training	-	-	-
$Y_k = \{y_{k_j}\}_{j=1}^{M_k}$	**the k^{th} group of targets**	-	-	-
$z_x = f_\theta(x)$	**input representation** from the input encoder f_θ	image embedding	image embedding	image embedding
$z_y = f_\phi(y)$	**target representation** from the target encoder f_ϕ	category embedding	text embedding	image embedding
$\hat{z}_y = f_\psi(z_x, t_x, t_y)$	**target prediction** from the decoder f_ψ	predicted embedding	predicted embedding	predicted embedding
$\boldsymbol{z}_x = f_\theta(\{x_i\}_{i=1}^{N})$	**input representation** from the input encoder f_θ	-	-	-
$\boldsymbol{z}_y^k = f_{\phi_k}(Y^k)$	**the k^{th} group target representation** from the target encoder f_{ϕ_k}	-	-	-
$\hat{\boldsymbol{z}}_y^k = f_{\psi_k}(\boldsymbol{z}_x, t_x, t_y)$	**the k^{th} group target prediction** from the decoder f_{ψ_k}	-	-	-
$\hat{P}(z_y\|\hat{z}_y)$	**approximated target posterior** given the prediction $\hat{z}_y$	Boltzmann	Boltzmann	Boltzmann / Gaussian
$\hat{P}_k(\boldsymbol{z}_y^k\|\hat{\boldsymbol{z}}_y^k)$	**approximated target posterior** given the prediction $\hat{\boldsymbol{z}}_y^k$	-	-	-
$H\big(p(z_y\|t_y)\big)$	**regularization term** to avoid representation collapse of z_y	negative categories	negative texts	negative samples / stop gradient / decorrelation
$H\Big(p\big(\{\boldsymbol{z}_y^k\}_{k=1}^{K}\|\boldsymbol{t}_y\big)\Big)$	**regularization term** to avoid representation collapse of $\{\boldsymbol{z}_y^k\}_{k=1}^{K}$	-	-	-

Table 6. Notation used in this paper. For single-input single-target pre-training, we also list the typical choices in different pre-training paradigm for each notation.

expectation because they should be deterministic once s, t_x, t_y and model parameters are given. The right-hand side of Eq. (7) is a lower bound of the actual mutual information and will be equal to it if and only if the estimated distribution $\hat{P}(z_y \mid \hat{z}_y)$ matches the real distribution $p(z_y \mid z_x, t_x, t_y)$. We note that because z_y should be a deterministic feature given z_x, t_x, t_y during training, equality can be achieved when the decoder predicts the target representation precisely. So we have

$$\begin{aligned} I(z_x;z_y \mid t_x,t_y) = \sup_{f_\psi}\ & \underbrace{\mathbb{E}_{p(t_y)}\Big[H\big(p(z_y(\phi)\mid t_y)\big)\Big]}_{\text{regularization term to avoid collapse}} \\ & + \underbrace{\mathop{\mathbb{E}}_{p(s,t_x,t_y)}\Big[\log \hat{P}\big(z_y(\phi)\mid \hat{z}_y(\theta,\psi)\big)\Big]}_{\text{prediction term for target representation}}. \end{aligned} \quad (8)$$

We usually deal with the regularization term in an implicit manner, such as introducing negative samples or stopping gradient to the target encoder. Therefore, the prediction term presents the loss function to be optimized in practice.

A.2. Multi-input Multi-target Pre-training

To derive the multi-input multi-target pre-training, we extend the input and the target to a set of N inputs $X = \{x_i\}_{i=1}^N$ and M targets $Y = \{y_j\}_{j=1}^M$. The set of targets are split into K non-overlapping groups as $Y_m \cap Y_{n\neq m} = \varnothing, \cup_{k=1}^K Y_k = Y$. The input representations and target representations are $\boldsymbol{z}_x = f_\theta(\{x_i\}_{i=1}^N)$ and $\boldsymbol{z}_y^k = f_{\phi_k}(Y_k)$, respectively. The mutual information is computed between $\boldsymbol{z}_x$ and $\{\boldsymbol{z}_y^k\}_{k=1}^K$ given the set of input transforms $\boldsymbol{t}_x$ and target transforms $\boldsymbol{t}_y$:

$$\max I(\boldsymbol{z}_x; \{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y). \quad (9)$$

Similar to Eq. (6), we can expand the mutual information as

$$\begin{aligned}
& I(\boldsymbol{z}_x; \{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y) \\
&= \int p(\boldsymbol{t}_x,\boldsymbol{t}_y) \int \Big[p(\boldsymbol{z}_x, \{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\cdot \\
&\qquad \log \frac{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)}{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)}\Big] d\boldsymbol{z}_x d\{\boldsymbol{z}_y^k\}_{k=1}^K d\boldsymbol{t}_x d\boldsymbol{t}_y \\
&= \int p(\boldsymbol{t}_x,\boldsymbol{t}_y) \int \Big[p(\boldsymbol{z}_x, \{\boldsymbol{z}_y^k\}_{k=1}^K \mid s, \boldsymbol{t}_x, \boldsymbol{t}_y) p(s \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\cdot \\
&\qquad \log \frac{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)}{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)}\Big] d\boldsymbol{z}_x d\{\boldsymbol{z}_y^k\}_{k=1}^K d\boldsymbol{t}_x d\boldsymbol{t}_y ds \\
&= \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x)}\Big[\int p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid Y)\cdot \\
&\qquad \log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y) d\{\boldsymbol{z}_y^k\}_{k=1}^K\Big] \\
&\quad - \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^K)}\Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\Big] \\
&= \sum_{k=1}^K \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x)}\Big[\int p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid Y)\cdot \\
&\qquad \log p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1}) d\{\boldsymbol{z}_y^k\}_{k=1}^K\Big] \\
&\quad - \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^K)}\Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\Big] \\
&= \sum_{k=1}^K \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x)}\Big[\int p(\{\boldsymbol{z}_y^i\}_{i=1}^{k-1} \mid Y) p(\boldsymbol{z}_y^k \mid Y)\cdot \\
&\qquad \underbrace{\int p(\{\boldsymbol{z}_y^i\}_{i=k+1}^{K} \mid Y) d\{\boldsymbol{z}_y^i\}_{i=k+1}^{K}}_{\text{the integral is equal to 1}}\cdot \\
&\qquad \log p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1}) d\{\boldsymbol{z}_y^i\}_{i=1}^{k}\Big] \\
&\quad - \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^K)}\Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\Big] \\
&= \underbrace{- \sum_{k=1}^K \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y,z_x,\{z_y^i\}_{i=1}^{k-1})}\Big[H\Big(p(\boldsymbol{z}_y^k \mid Y_k), p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1})\Big)\Big]}_{\text{prediction term for target representations}} \\
&\quad \underbrace{+ \mathbb{E}_{p(\boldsymbol{t}_y)}\Big[H\Big(p(\{z_y^k\}_{k=1}^K \mid \boldsymbol{t}_y)\Big)\Big]}_{\text{regularization term to avoid collapse}},
\end{aligned} \quad (10)$$

where the fifth equality hold because the target representations are independent given targets, and $\{\boldsymbol{z}_y^i\}_{i=1}^{k-1} = \varnothing$ for $k = 1$.

During parameterization, we adopt different predictions $\hat{\boldsymbol{z}}_y^k = f_{\psi_k}(\boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)$ and distributions $\hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k)$ for different target groups. Then the mutual information can be converted into

$$\begin{aligned}
& I(\boldsymbol{z}_x; \{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y) \\
&= \int p(\boldsymbol{t}_x,\boldsymbol{t}_y) \int \Big[p(\boldsymbol{z}_x \mid \boldsymbol{t}_x, \boldsymbol{t}_y) p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)\cdot \\
&\qquad \log \frac{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)}{p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)}\Big] d\boldsymbol{z}_x d\{\boldsymbol{z}_y^k\}_{k=1}^K d\boldsymbol{t}_x d\boldsymbol{t}_y \\
&= \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x)}\Big[\int p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y)\cdot \\
&\qquad \log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y) d\{\boldsymbol{z}_y^k\}_{k=1}^K\Big] \\
&\quad - \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^K)}\Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^K \mid \boldsymbol{t}_x, \boldsymbol{t}_y)\Big]
\end{aligned}$$

$$
\begin{aligned}
&= \underbrace{\sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^i\}_{i=1}^{k-1})} \Bigg[\int p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1}) \cdot \log \frac{p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1})}{\hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k)} d\boldsymbol{z}_y^k \Bigg]}_{\text{KL Divergence} \geq 0} \\
&\quad + \sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^i\}_{i=1}^{k-1})} \Bigg[\int p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1}) \cdot \log \hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k) d\boldsymbol{z}_y^k \Bigg] \\
&\quad - \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^{K})} \Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_x, \boldsymbol{t}_y) \Big] \\
&\geq \sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^i\}_{i=1}^{k-1})} \Bigg[\int p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1}) \cdot \log \hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k) d\boldsymbol{z}_y^k \Bigg] \\
&\quad - \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^k\}_{k=1}^{K})} \Big[\log p(\{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_x, \boldsymbol{t}_y) \Big] \\
&= \sum_{k=1}^{K} \mathbb{E}_{p(\boldsymbol{t}_x,\boldsymbol{t}_y,\boldsymbol{z}_x,\{\boldsymbol{z}_y^i\}_{i=1}^{k})} \Bigg[\log \hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k) \cdot \underbrace{\int p(s \mid \boldsymbol{t}_x, \boldsymbol{t}_y, \boldsymbol{z}_x, \{\boldsymbol{z}_y^i\}_{i=1}^{k}) ds}_{\text{the integral is equal to 1}} \Bigg] \\
&\quad + \mathbb{E}_{p(\boldsymbol{t}_y)} \Bigg[p(\{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_y) \log p(\{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_y) \cdot \underbrace{\int p(\boldsymbol{z}_x, \boldsymbol{t}_x \mid \{\boldsymbol{z}_y^k\}_{k=1}^{K}, \boldsymbol{t}_y) d\boldsymbol{z}_x d\boldsymbol{t}_x}_{\text{the integral is equal to 1}} \, d\{\boldsymbol{z}_y^k\}_{k=1}^{K} \Bigg] \\
&= \underbrace{\sum_{k=1}^{K} \mathbb{E}_{p(s,\boldsymbol{t}_x,\boldsymbol{t}_y)} \Big[\log \hat{P}_k\Big(\boldsymbol{z}_y^k(\phi_k) \mid \hat{\boldsymbol{z}}_y^k(\theta, \psi_k)\Big) \Big]}_{\text{prediction term for target representations}} \\
&\quad + \underbrace{\mathbb{E}_{p(\boldsymbol{t}_y)} \Big[H(\{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_y) \Big]}_{\text{regularization term to avoid collapse}},
\end{aligned} \tag{11}
$$

where the fourth inequality holds because KL Divergence will not be less than 0 for every summation term. The equality can be achieved if and only if every $\hat{P}_k(\boldsymbol{z}_y^k \mid \hat{\boldsymbol{z}}_y^k)$ matches $p(\boldsymbol{z}_y^k \mid \boldsymbol{z}_x, \boldsymbol{t}_x, \boldsymbol{t}_y, \{\boldsymbol{z}_y^i\}_{i=1}^{k-1})$. Therefore, the mutual information for multi-input multi-target pre-training can be bounded by

$$
\begin{aligned}
&I\big(\boldsymbol{z}_x; \{\boldsymbol{z}_y^k\}_{k=1}^{K} \mid \boldsymbol{t}_x, \boldsymbol{t}_y\big) \\
&\geq \sup_{\{f_{\psi_k}\}_{k=1}^{K}} \underbrace{\mathop{\mathbb{E}}_{p(\boldsymbol{t}_y)} \Big[H\Big(p(\{\boldsymbol{z}_y^k(\phi_k)\}_{k=1}^{K} \mid \boldsymbol{t}_y)\Big) \Big]}_{\text{regularization term to avoid collapse}} \\
&\quad + \underbrace{\sum_{k=1}^{K} \mathop{\mathbb{E}}_{p(\boldsymbol{s},\boldsymbol{t}_x,\boldsymbol{t}_y)} \Big[\log \hat{P}_k\big(\boldsymbol{z}_y^k(\phi_k) \mid \hat{\boldsymbol{z}}_y^k(\theta, \psi_k)\big) \Big]}_{\text{prediction term for target representation}}.
\end{aligned} \tag{12}
$$

It's shown that different target groups are disentangled into a summation of prediction terms, so we can optimize each target objective independently.

B. Experiment Details

B.1. Pre-training Settings

We utilize InternImage-H as image encoder in Sec 4.1 for large model pre-training and ViT-B/16 as that in other experiments for ablation study and fair comparison. For image-text dataset (*e.g.,* YFCC-15M [68]), a 12-layer Transformer (with the same network architecture as BERT-Base [22]) is utilized as text target encoder. For image classification dataset (*e.g.,* ImageNet [21]), we directly use the linear classifier weight as category embedding target. We employ 4-layer Transformer as decoder for image representation target, and Attention Pooling as that for category embedding or text global feature. Detailed hyper-parameters for pre-training InternImage-H and ViT-B are listed in Tab. 7.

Dynamic weighting is used to balance the weights of self-supervised loss (L_{SSP}) and supervised/weakly-supervised loss (L_{SP}). The overall training loss can be expressed as

$$
L = L_{\text{SSP}} + \lambda L_{\text{SP}}, \tag{13}
$$

where λ is the balance loss weight. Because the loss behavior changes dramatically during training, it's sub-optimal to set a static weight. We propose to set λ dynamically according to the loss gradients. Specifically, we compute the exponential moving average of gradient norm that each loss back-propagates to input features, denoted as $\bar{g}_{\text{uni-modal}}$ and $\bar{g}_{\text{multi-modal}}$. Then λ is set as $\gamma \cdot \bar{g}_{\text{uni-modal}} / \bar{g}_{\text{multi-modal}}$, where γ controls the gradient ratio between two loss terms. We find this strategy to work well in practice ($\gamma = 1$ by default).

B.2. Tranfer Settings of InternImage-H

We strictly follow [79] for the transfer settings of InternImage-H on ImageNet-1k, COCO, LVIS and ADE20k. We briefly summarize the settings below.

ImageNet-1k. For ImageNet classification, the pre-trained InternImage-H is fine-tuned on ImageNet-1k for 30 epochs.

COCO. For COCO object detection, we double the parameters of pre-trained InternImage-H via the composite techniques [45]. Then it is fine-tuned with the DINO [95] detector on Objects365 [57] and COCO datasets one after another for 26 epochs and 12 epochs.

Hyper-parameters	ViT-B/16	InternImage-H
Image-to-image decoder layers	4	
Image-to-image decoder hidden size	768	1024
Image-to-image decoder FFN hidden size	3072	4096
Image-to-image decoder attention heads	16	
Attention pooling input size	768	1024
Attention pooling output size	768	
Attention pooling attention heads	16	
Data augment	RandomResizedCrop RandomHorizontalFlip ColorJitter RandomGrayscale GaussianBlur Solarize	
Mask strategy	Blockwise mask	
Mask ratio	50%	
Input resolution	224 × 224	192 × 192

Training epochs	1600(ImageNet)
	138(YFCC)
		30

Batch size	4096	40000
Adam β	(0.9, 0.95)	
Peak learning rate	1.0×10^{-3}	
Learning rate schedule	cosine	

Warmup epochs	40(ImageNet)
	3.5(YFCC)
		1

Weight decay	0.1	
EMA coeff	0.995	
EMA schedule	cosine	
Label smoothing	0.1	
Stock. depth	0.1 (linear)	0.2 (uniform)

Table 7. Hyper-parameters for pre-training.

LVIS. For LVIS long-tailed object detection, we double the parameters of pre-trained InternImage-H via the composite techniques [45]. Then it is fine-tuned with the DINO [95] detector on Objects365 [57] and LVIS datasets one after another for 26 epochs and 12 epochs.

ADE20k. For ADE20k semantic segmentation, we fine-tune InternImage-H with Mask2Former [18], and adopt the same settings in [17, 78].

B.3. Transfer Settings of ViT-B/16

ImageNet-1k. The detailed fine-tuning and linear classification settings of ViT-B/16 on ImageNet-1k are listed in Tab. 8 and Tab. 9.

COCO and LVIS. We utilize ViTDet [43] for object detection. By default, the fine-tuning schedule is set to 100 epochs for both COCO and LVIS datasets. For the ablation study, we use a short schedule of 25 epochs. Detailed hyper-parameters are listed in Tab. 10 and Tab. 11.

ADE20k. Following [5, 30, 67], we employ UperNet [83] as the segmentation network. We use the implementation in MMSegmentation [19]. Detailed hyper-parameters are listed in Tab. 12.

Hyper-parameters	Value
Erasing prob.	0.25
Rand augment	9/0.5
Mixup prob.	0.8
Cutmix prob.	1.0
Input resolution	224 × 224
Finetuning epochs	100
Batch size	1024
Adam β	(0.9, 0.999)
Peak learning rate	2.0×10^{-3}
Learning rate schedule	cosine
Warmup epochs	5
Weight decay	0.1
Layer-wise learning rate decay	0.65
Label smoothing	0.1
Stock. depth	0.1

Table 8. Hyper-parameters of ViT-B for ImageNet finetuning.

Hyper-parameters	Value
Data augment	RandomResizedCrop RandomHorizontalFlip
Input resolution	224 × 224
Training epochs	90
Batch size	16384
Optimizer	LARS
Peak learning rate	3.2
Learning rate schedule	cosine
Warmup epochs	10
Weight decay	0.0

Table 9. Hyper-parameters of ViT-B for ImageNet linear probing.

Hyper-parameters	Value
Data augment	large scale jittor
Input resolution	1024 × 1024
Finetuning epochs	100
Batch size	64
Adam β	(0.9, 0.999)
Peak learning rate	1.0×10^{-4}
Learning rate schedule	step
Warmup length	250 iters
Weight decay	0.1
Stock. depth	0.1
Relative positional embeddings	✓

Table 10. Hyper-parameters of ViT-B for COCO detection.

Hyper-parameters	Value
Data augment	large scale jittor
Input resolution	1024 × 1024
Finetuning epochs	100
Batch size	64
Adam β	(0.9, 0.999)
Peak learning rate	2.0×10^{-4}
Learning rate schedule	step
Warmup length	250 iters
Weight decay	0.1
Stock. depth	0.1
Relative positional embeddings	✓

Table 11. Hyper-parameters of ViT-B for LVIS detection.

Hyper-parameters	Value
Data augment	RandomCrop RandomFlip PhotoMetricDistortion
Input resolution	512 × 512
Finetuning length	160k iters
Batch size	16
Adam β	(0.9, 0.999)
Peak learning rate	1.0×10^{-4}
Learning rate schedule	linear
Warmup length	1500 iters
Weight decay	0.05
Stock. depth	0.1
Relative positional embeddings	✓

Table 12. Hyper-parameters of ViT-B for ADE20k semantic segmentatioin.

Gradient Ratio γ	0.2	0.5	1.0	2.0	5.0
ImageNet Top1	83.1	83.2	**83.3**	82.8	82.5
COCO AP^{box}	50.2	**50.5**	**50.5**	48.9	47.6

Table 13. Ablation study of gradient ratio γ.

B.4. More Experiments

Ablation Study on Gradient Ratio γ. The gradient ratio γ is used in dynamic weighting (see Eq. (13) in Appendix B.1). We ablate the choice of γ from $\{0.2, 0.5, 1.0, 2.0, 5.0\}$ in Tab. 13. These models are pre-trained on ImageNet-1k for 100 epochs. Then they are fine-tuned on ImageNet-1k classification and COCO object detection. The fine-tuning schedules for ImageNet-1k and COCO are set to 100 epochs and 25 epochs respectively. As shown in Tab. 13, $\gamma = 0.2, 0.5, 1.0$ works quite well in both classification and detection. We choose $\gamma = 1.0$ as our default setting for its simplicity.

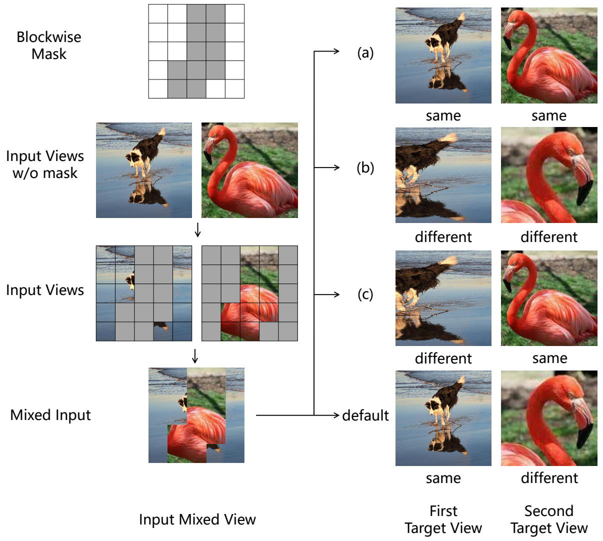

Figure 3. Illustration of four design choices of target views.

	First Target View	Second Target View	ImageNet Top1†	COCO AP^{box}
(a)	same	same	77.2	48.6
(b)	different	different	78.5	**49.8**
(c)	different	same	78.8	49.2
default	**same**	**different**	**79.1**	49.5

Table 14. Ablation study of target views. † ImageNet fine-tuning is early-stopped at 20 epochs which we found consistent with the final performance in practice.

Ablation Study on Target Views. Our M3I Pre-training consists of two target image views during the multi-input multi-target pre-training. Two input views of different images are mixed with a shared blockwise mask. As shown in Fig. 3, the visible part of the blockwise mask is filled with an augmented view of the first image, and the masked part is filled with an augmented view of the second image. The first target view and second target view are not permutable. We ablate the choices of these two target image views (either the same or different from the input image view) in Tab. 14. These models are pre-trained on ImageNet-1k without labels (*i.e.,* they only have image representation target and do not have the category embedding target) for 200 epochs. Then, they are fine-tuned on ImageNet-1k classification and COCO object detection. The fine-tuning schedule is set to 100 epochs and 25 epochs respectively. Our default setting works best in ImageNet classification. Although (b) perform slightly better than our default setting in COCO detection, the pre-training process of it is quite unstable (FP16 loss scale is quite unstable), thus we do not choose it as our default setting.

Experiment Results on Image-Text Retrieval. Following the setting of BEiT-3 [78], our M3I Pre-training achieves 89.1 R@1 on Flickr30K zero-shot retrieval task, which is better than BEiT-3 (88.2 R@1).

Model	Pre-train Epochs	ImageNet Top1
ViT-B/16	400	83.9
ViT-L/16	400	86.0

Table 15. Comparison of ViT-B/16 and ViT-L/16. The models are pre-trained on ImageNet for 400 epochs and fine-tuned on ImageNet classification for 100 epochs.

Experiment Results on ViT-L/16. We utilize our M3I to pre-train ViT-L/16 on ImageNet for 400 epochs and compare it with ViT-B/16. As shown in Tab. 15, ViT-L/16 achieves 86.0 Top1 accuracy on ImageNet fine-tuning.