

Stargan

✓ 论文出发点

✎ Cycle-Gan中如果要生成一种效果，需要训练一组配对的G和D (2+2)

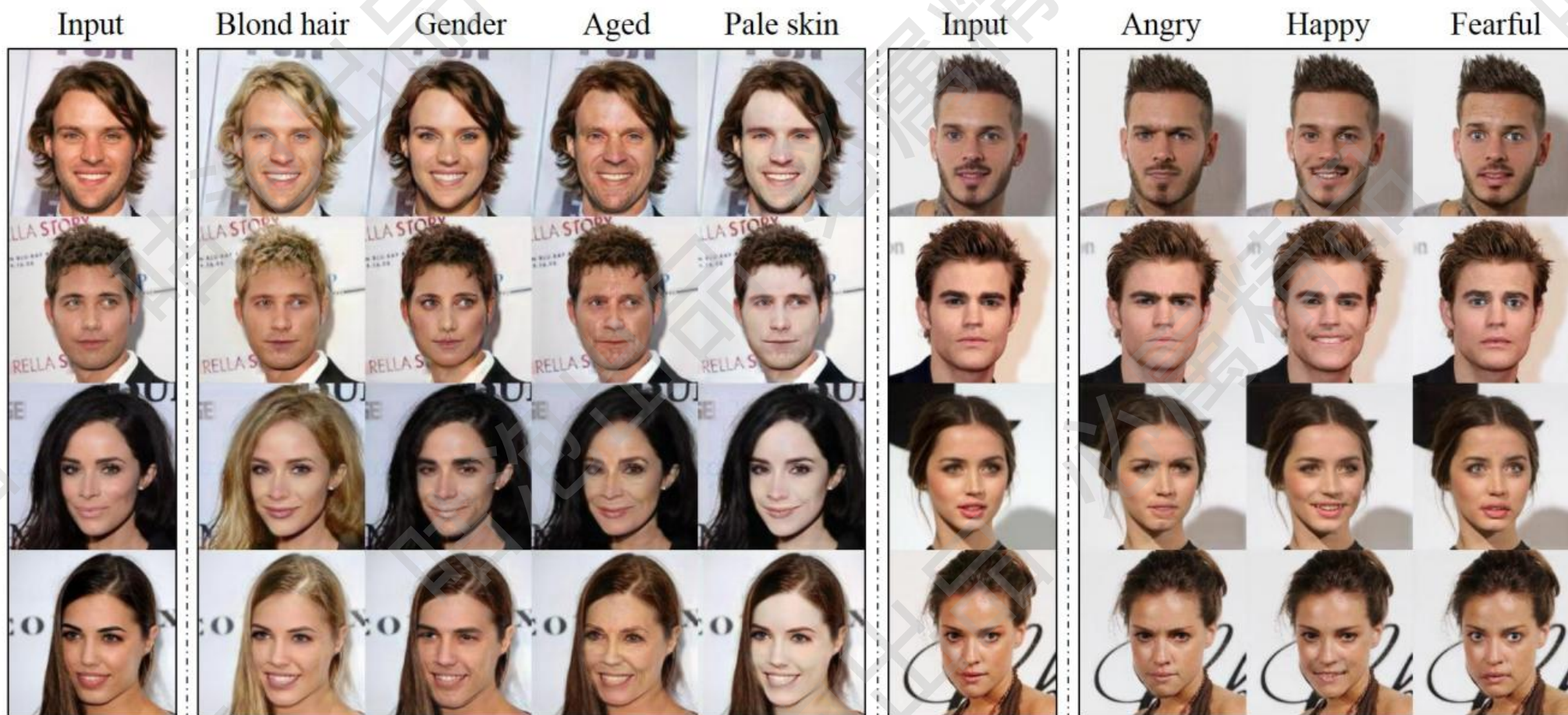
✎ 如果我想DIY，大鼻子，大眼睛，大嘴巴，大耳朵。。。

✎ 能不能一个模型搞定这些事呢？显然是可以的，明星登场

✎ 现在已经两代版本了v1和v2，源码基于V2讲解及演示

Stargan

✓ 先来看看要干啥



Stargan

✓ star为何意?

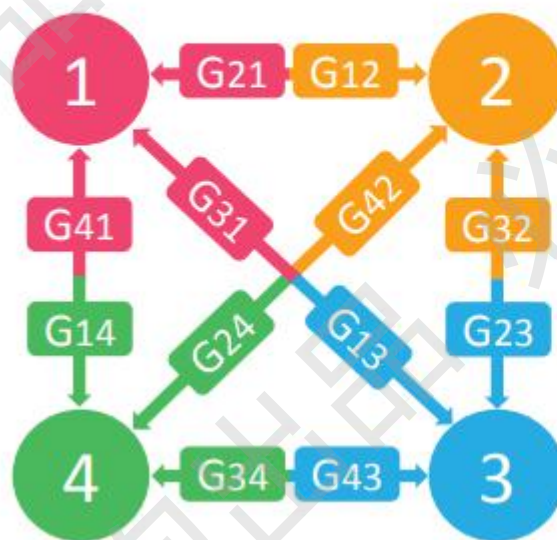
✎ 如果用CycleGan, 需要训练出 $K*(K-1)$ 个G网络, 如果star呢?

✎ 一个G就够了!

✎ 看看样子是不是很像一个star

✎ 如何做到只用一个G呢?

(a) Cross-domain models



(b) StarGAN



Stargan

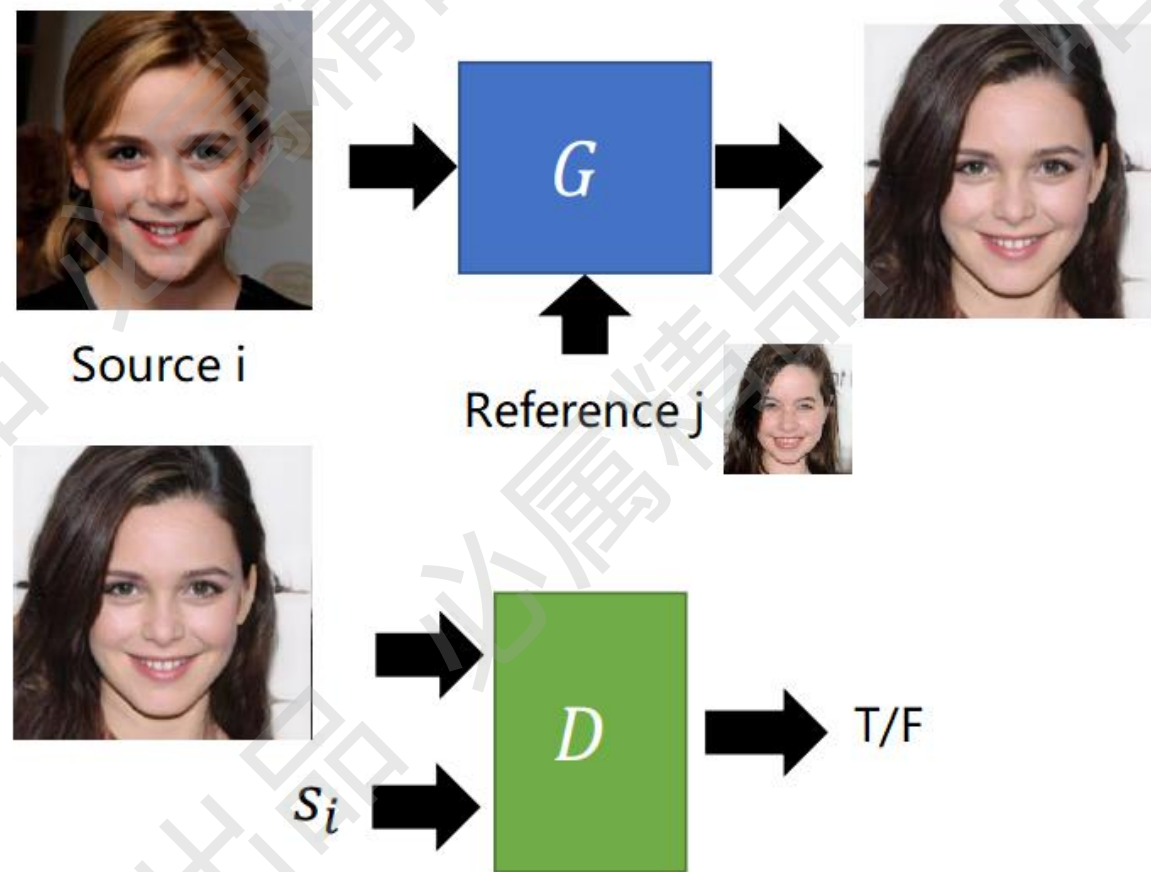
✓ 基本思路

✎ G额外接受了编码信息 (one-hot)

✎ D也同样接受, 只需一个D与G

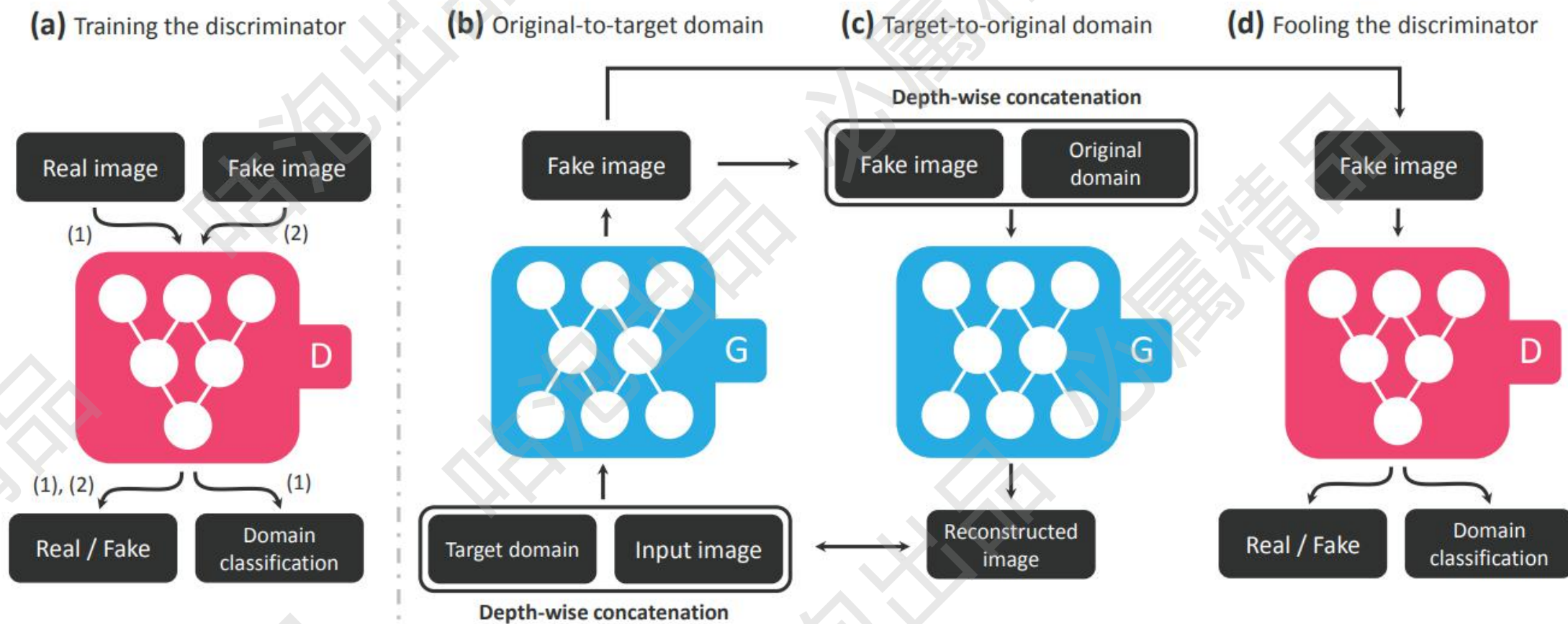
✎ 整体流程类似Cycle-Gan

✎ 现在一个G就可以生成各种效果



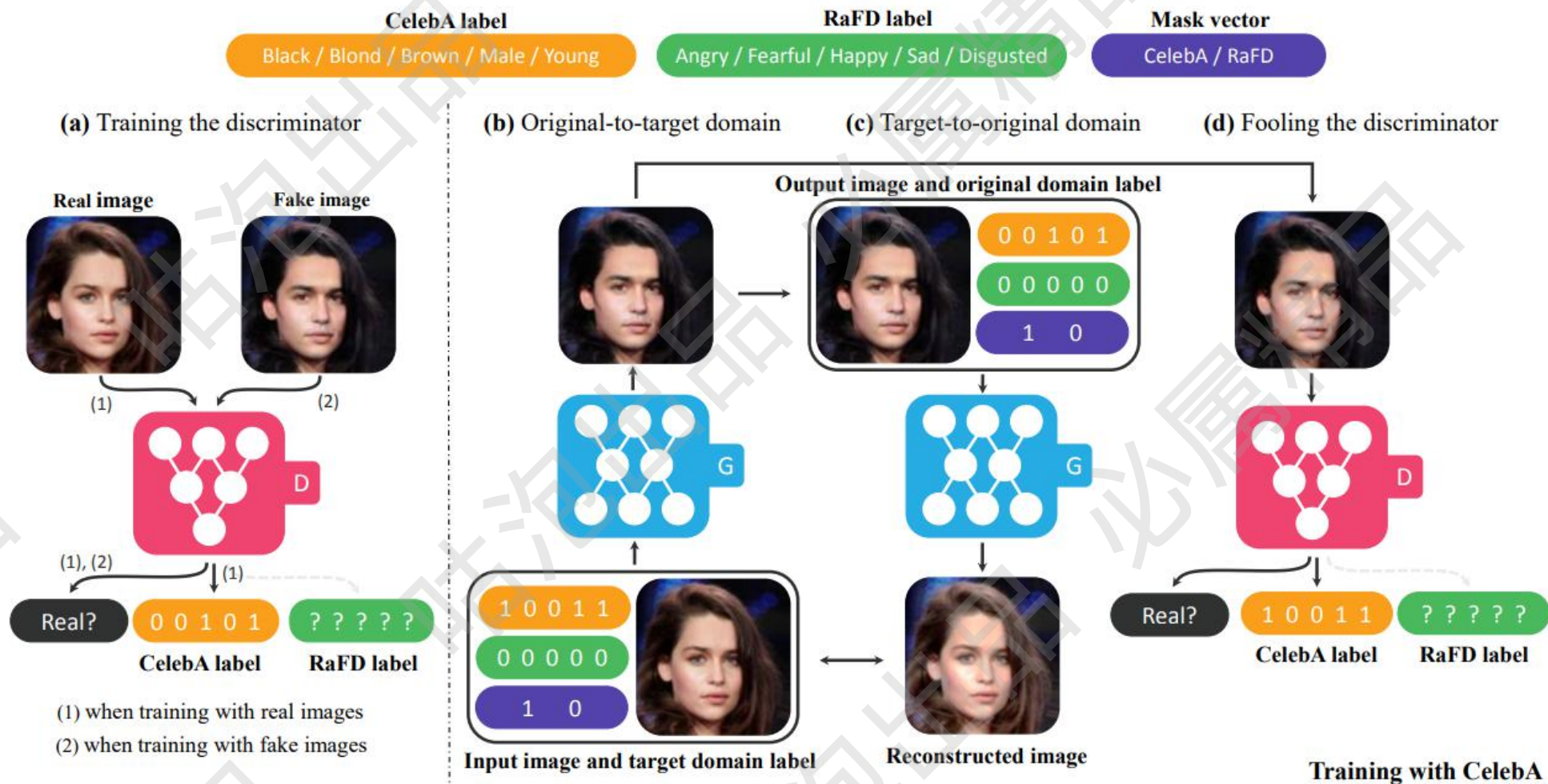
Stargan

✓ 整体流程



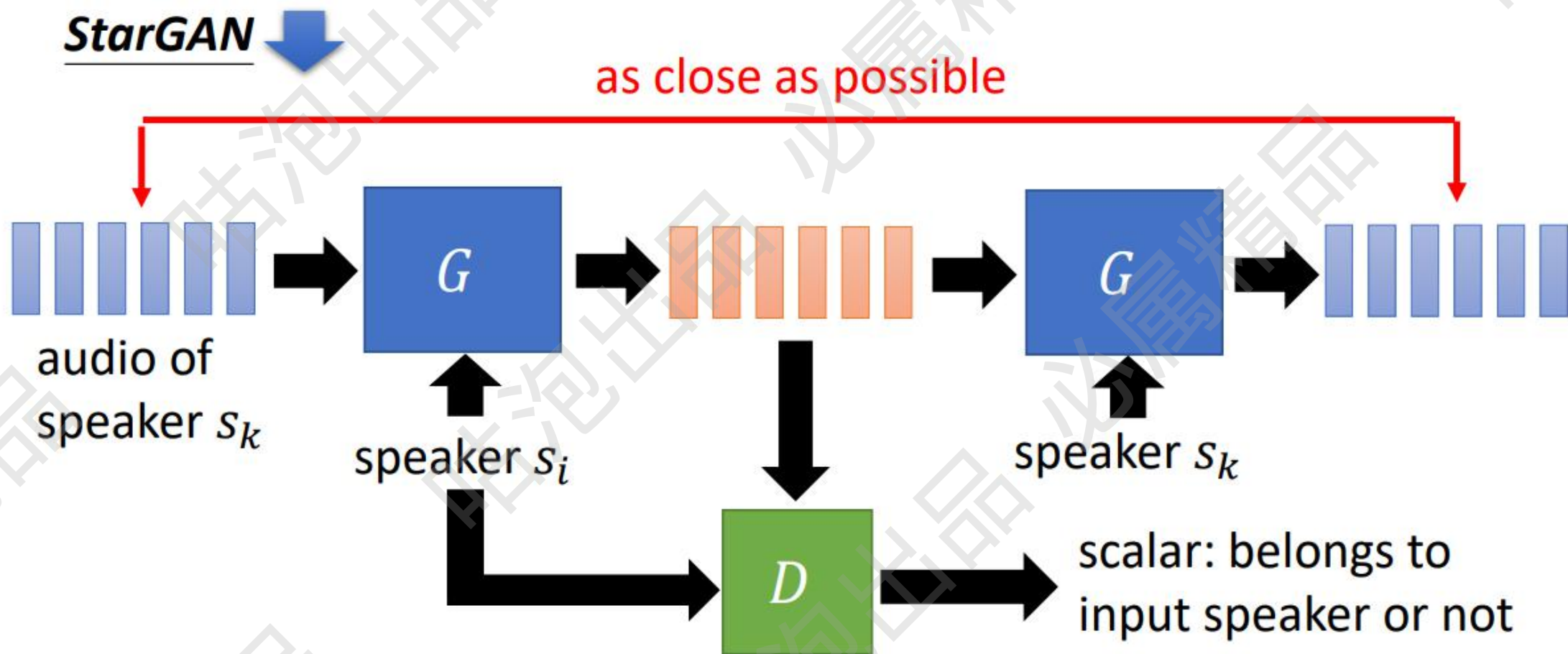
Stargan

✓ 整体流程



Stargan

✓ 变音器是怎么来的呢?



Stargan

✓ V2版本

✎ one-hot向量真的合适吗？想一想NLP里面现在还用one-hot吗？

✎ 每一种风格能不能编码成为一个向量呢？

✎ V2版本引入了几个新哥们，整体框架还是star结构

✎ 多了Mapping network与Style encoder两个新成员

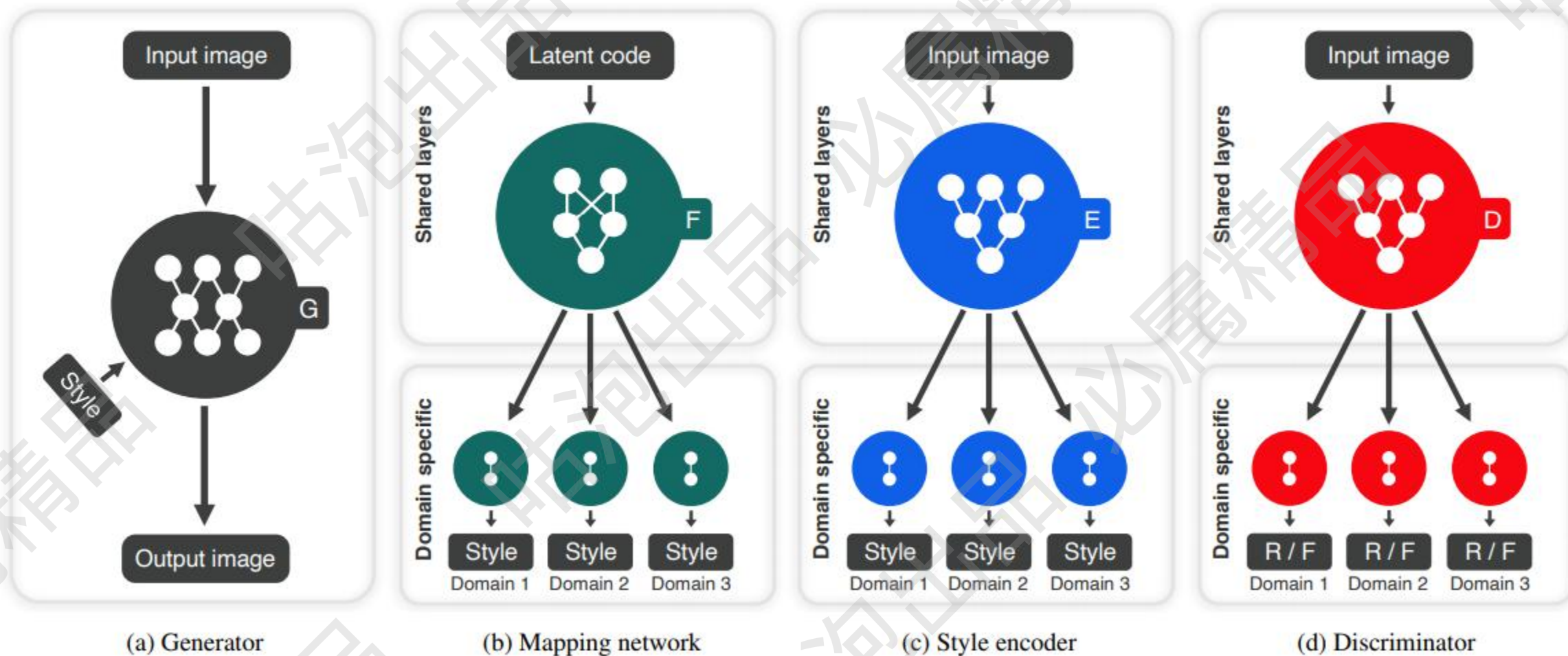
Stargan

✓ 先看效果



Stargan

✓ 整体网络模型



Stargan

✓ Style reconstruction (编码器训练)

✎ 风格编码器损失函数: $\mathcal{L}_{sty} = \mathbb{E}_{\mathbf{x}, \tilde{y}, \mathbf{z}} [||\tilde{\mathbf{s}} - E_{\tilde{y}}(G(\mathbf{x}, \tilde{\mathbf{s}}))||_1]$

✎ 其中 $\tilde{\mathbf{s}}$ 是由mapping网络所得到的结果

✎ 相当于先给G一组风格向量，让它去生成，再对结果的结果进行编码

✎ 让编码后得到的风格向量跟输入的mapping网络向量越接近越好

Stargan

✓ Style diversification (多样化训练)

✎ 结果得生成一些, 多样化损失: $\mathcal{L}_{ds} = \mathbb{E}_{\mathbf{x}, \tilde{y}, \mathbf{z}_1, \mathbf{z}_2} [\|G(\mathbf{x}, \tilde{\mathbf{s}}_1) - G(\mathbf{x}, \tilde{\mathbf{s}}_2)\|_1]$

✎ 两组随机向量 $\mathbf{z}_1, \mathbf{z}_2$ 通过mapping网络可以得到 \mathbf{s}_1 与 \mathbf{s}_2

$$\tilde{\mathbf{s}}_i = F_{\tilde{y}}(\mathbf{z}_i) \text{ for } i \in \{1, 2\}$$

✎ 为了结果丰富, 要使得当下计算结果越大越好, 求损失时去负号

Stargan

✓ cycle loss(不能人变得都认不出来吧)

✎ 转换的只是特色，主体得保留： $\mathcal{L}_{cyc} = \mathbb{E}_{\mathbf{x}, y, \tilde{y}, \mathbf{z}} [\|\mathbf{x} - G(G(\mathbf{x}, \tilde{\mathbf{s}}), \hat{\mathbf{s}})\|_1]$

✎ 其中： $\hat{\mathbf{s}} = E_y(\mathbf{x})$ 也就是原始图像本身的风格向量

✎ 训练时可以保证图像不至于变得太离谱

✎ 这个损失函数其实跟cycle Gan基本就是一样