2. Prepare figures, tables & captions

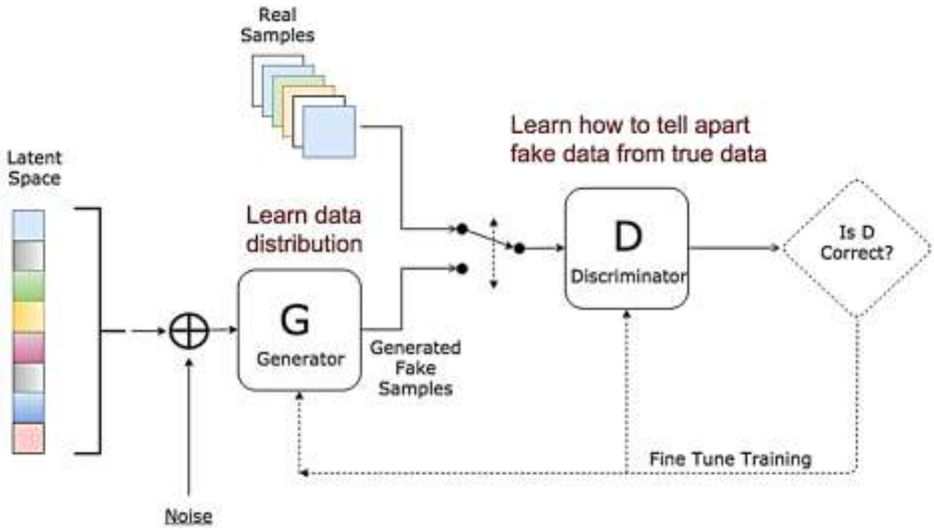

Figure 1: Generative Adversarial Network (GAN)

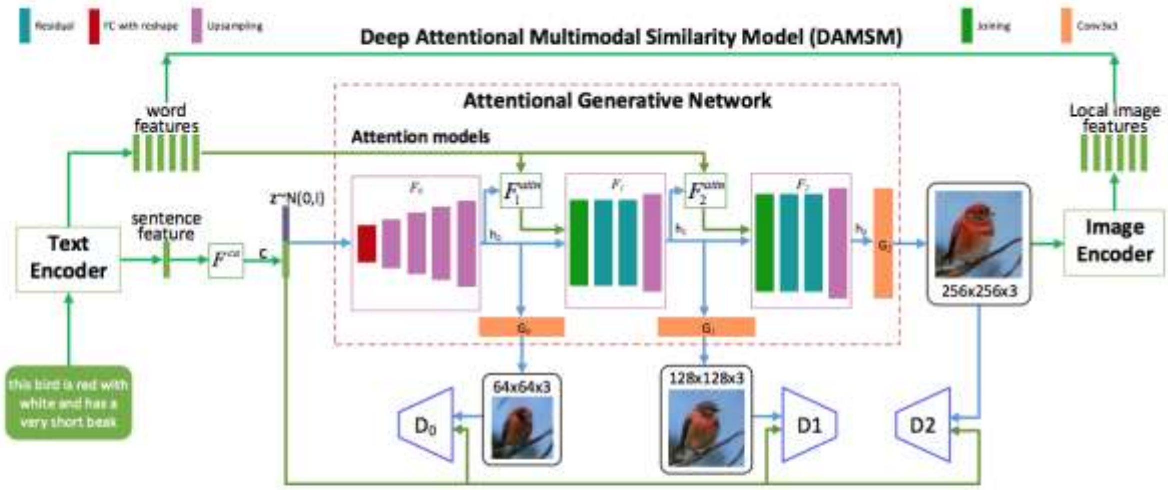

Figure 2: AttnGAN

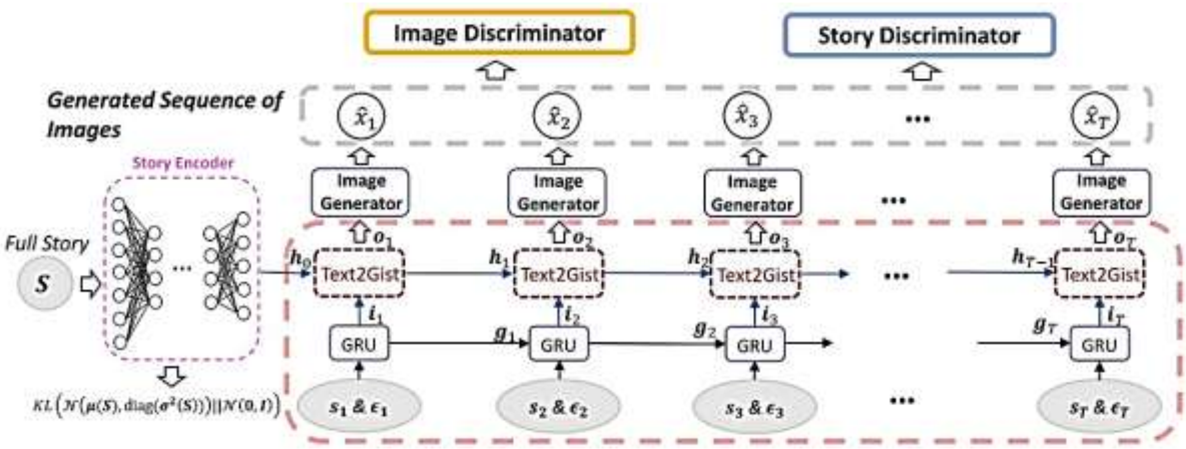

Figure 3: StoryGAN

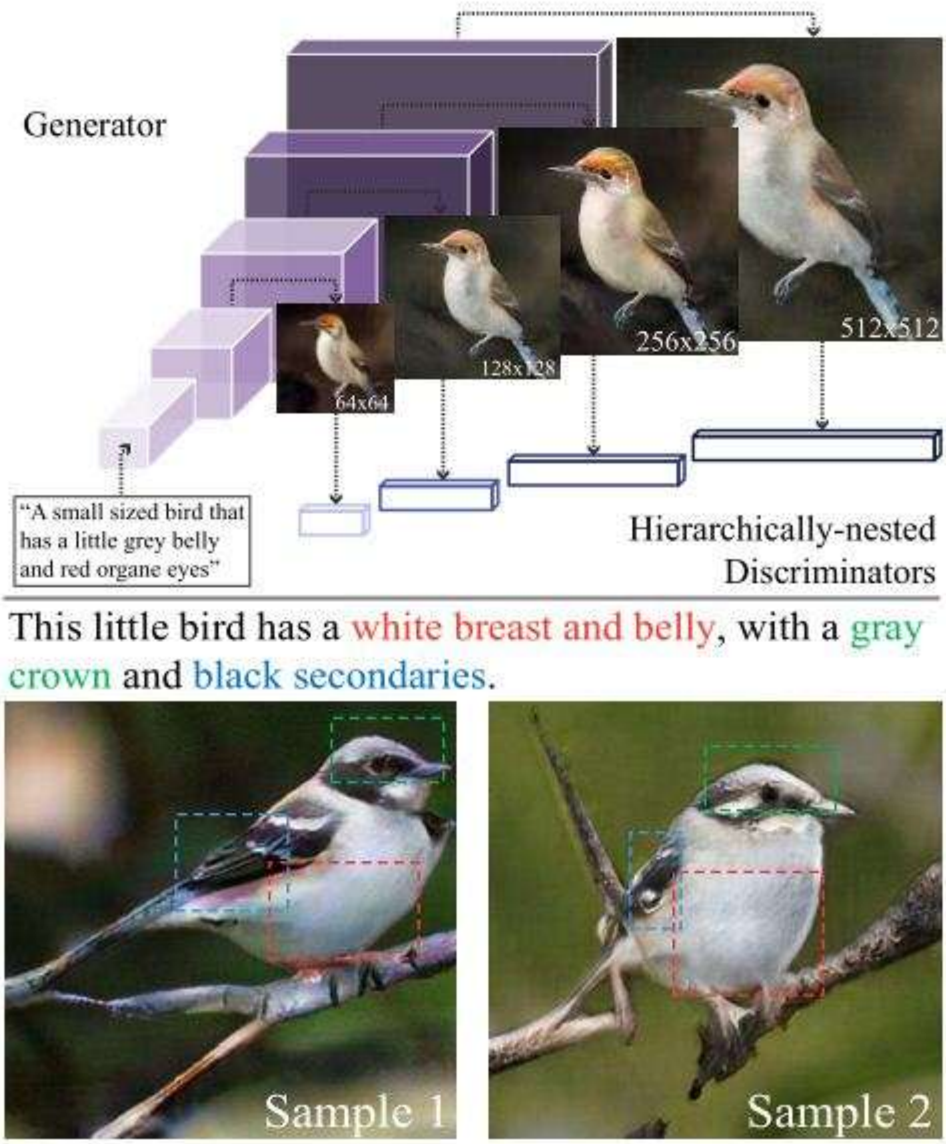

Figure 1: Top: Overview of our hierarchically-nested adversarial network, which produces side output images with growing resolutions. Each side output is associated with a discriminator. Bottom: Two test sample results where fine-grained details are highlighted.

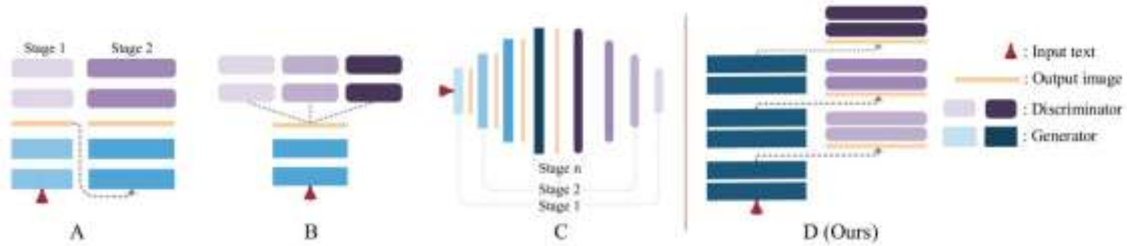

Figure 2: Overviews of some typical GAN frameworks. **A** uses multi-stage GANs [46, 6]. **B** uses multiple discriminators with one generator [9, 28]. **C** progressively trains symmetric discriminators and generators [16, 12]. **A** and **C** can be viewed as decomposing high-resolution tasks to multi-stage low-to-high resolution tasks. **D** is our proposed framework that uses a single-stream generator with hierarchically-nested discriminators trained end-to-end.

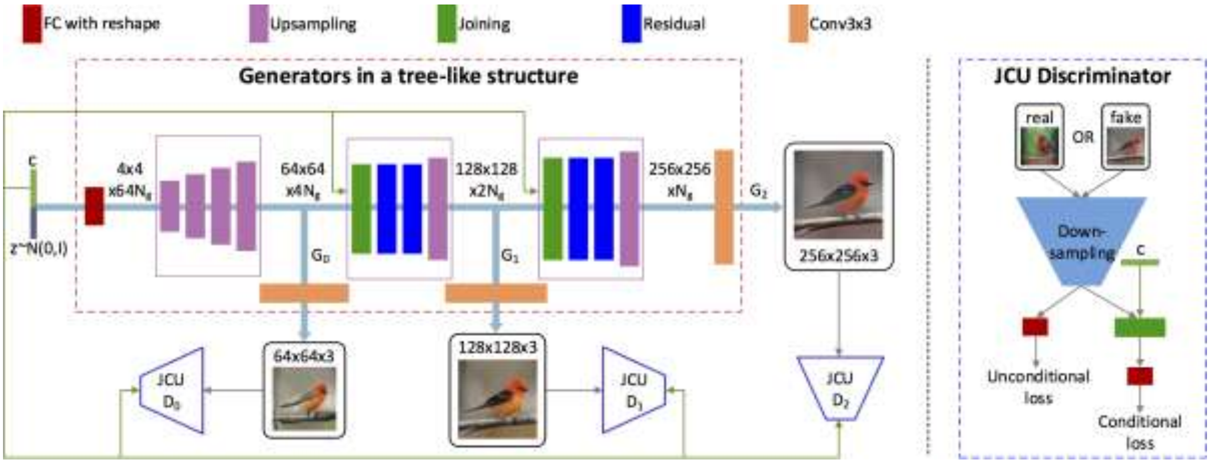

Figure 4: StackGAN++

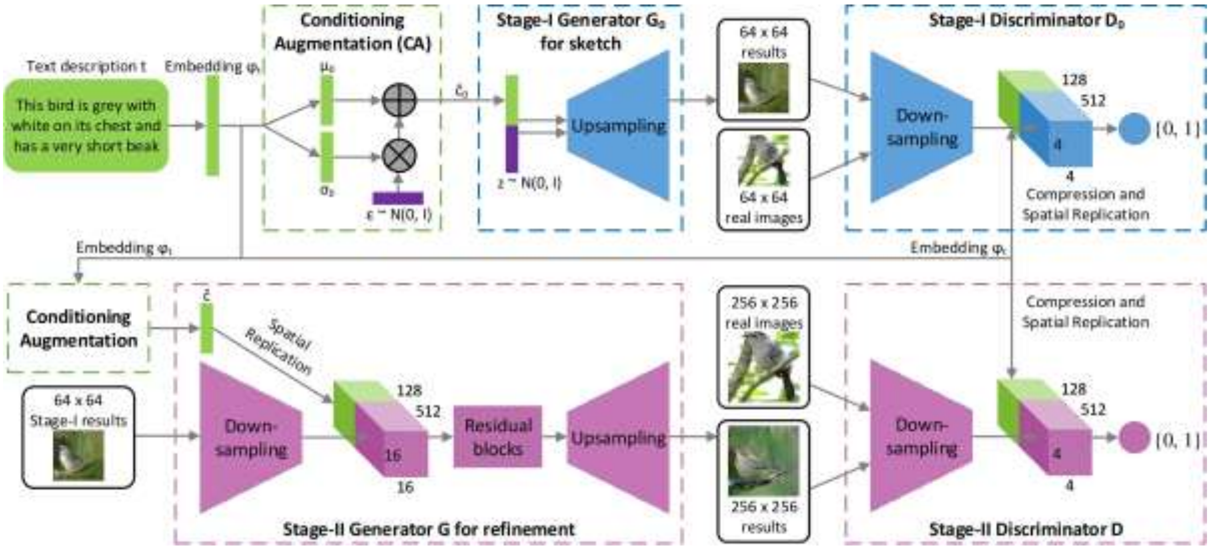

Figure 5: StackGAN

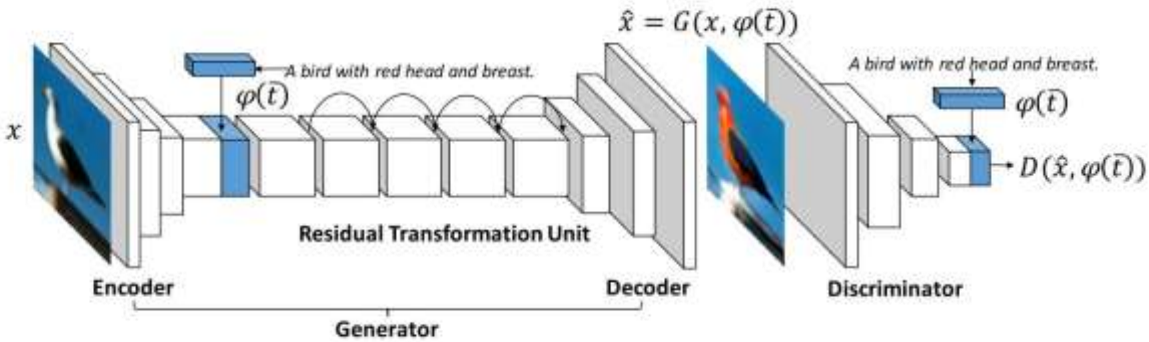

Figure 2. Network architecture of our proposed model. It consists of a generator network and a discriminator network. The generator has an encoder-decoder architecture and synthesizes images conditioned on both images and text embeddings. The discriminator performs the discriminative task conditioned on text embeddings.

Figure 6: SemanticGAN

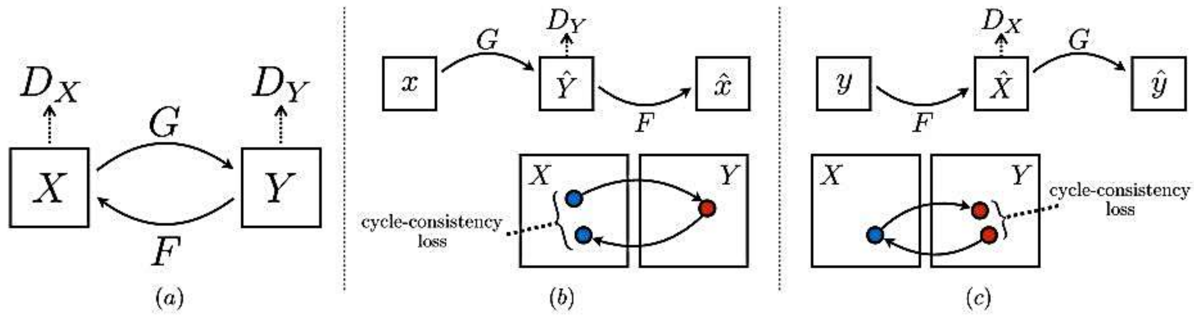

Figure 3: (a) Our model contains two mapping functions $G : X \to Y$ and $F : Y \to X$, and associated adversarial discriminators D_Y and D_X. D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss: $x \to G(x) \to F(G(x)) \approx x$, and (c) backward cycle-consistency loss: $y \to F(y) \to G(F(y)) \approx y$

Figure 7: CycleGAN

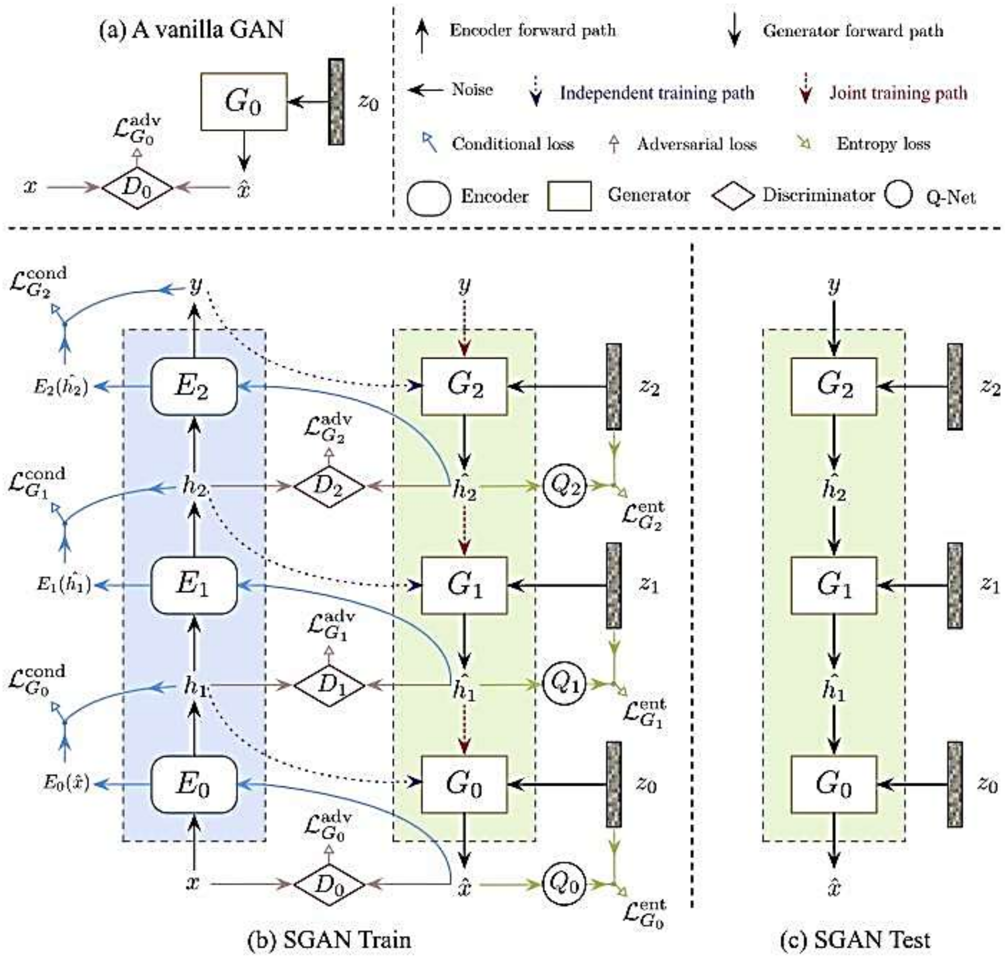

Figure 1: **An overview of SGAN.** (a) The original GAN in [17]. (b) The workflow of training SGAN, where each generator G_i tries to generate plausible features that can fool the corresponding representation discriminator D_i. Each generator receives conditional input from encoders in the independent training stage, and from the upper generators in the joint training stage. (c) New images can be sampled from SGAN (during test time) by feeding random noise to each generator G_i.

Figure 8: StackedGAN

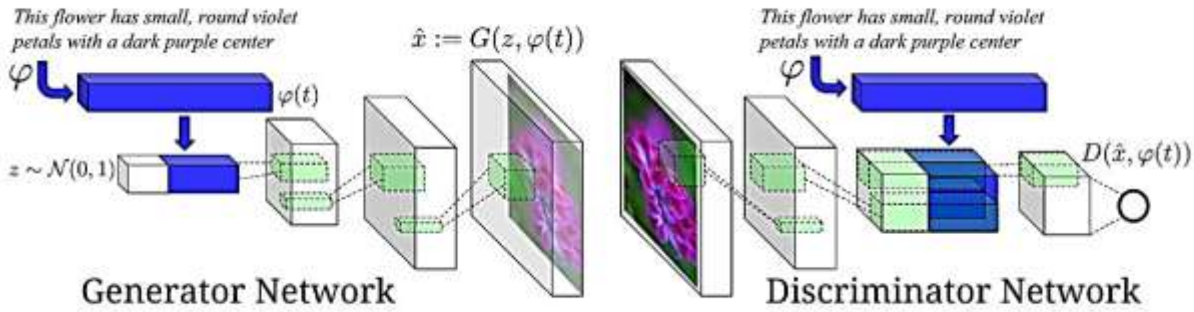

Figure 2. Our text-conditional convolutional GAN architecture. Text encoding $\varphi(t)$ is used by both generator and discriminator. It is projected to a lower-dimensions and depth concatenated with image feature maps for further stages of convolutional processing.

Figure 9: Reed GAN

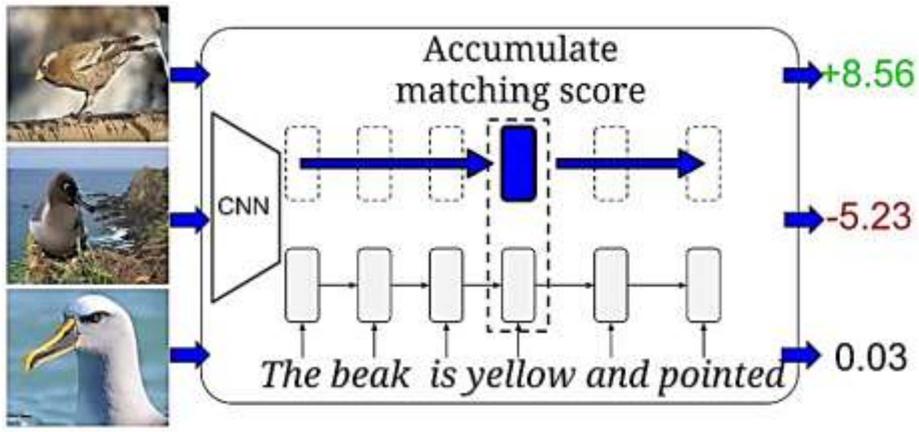

Figure 1: Our model learns a scoring function between images and text descriptions. A word-based LSTM is shown here, but we also evaluate several alternative models.

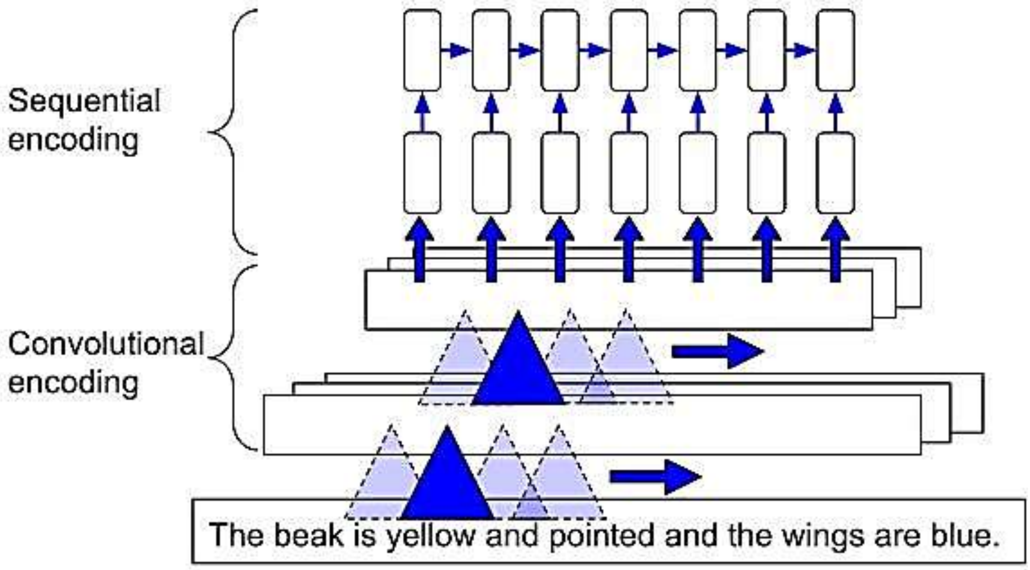

Figure 2: Our proposed convolutional-recurrent net.