**IMPORTANT FEATURES OF BigGAN**

### **Orthogonal Regularization**

#### **Purpose:**

* Prevent **mode collapse**, where the generator produces limited diversity in outputs (e.g., the same few images repeatedly).
* Encourage **orthogonality** in weight matrices, which helps the generator maintain diverse outputs.

#### **How It Works:**

* Orthogonal regularization penalizes the deviation of weight matrices from being orthogonal.
* **Orthogonal matrices** have the property that their rows (or columns) are mutually orthogonal (dot product = 0). This property ensures that the network learns independent and diverse features.

#### **Why It Helps:**

1. Encourages the generator to explore diverse modes in the data distribution.
2. Reduces overfitting and prevents degenerate solutions.
3. Helps stabilize training, especially in high-dimensional latent spaces.

### **2. Spectral Normalization**

#### **Purpose:**

* Stabilize training by constraining the **Lipschitz constant** of the network.
* Prevent the discriminator from becoming overly strong, which can hinder the generator's learning.

#### **How It Works:**

* Spectral Normalization controls the largest singular value (σmax\sigma\_{max}σmax​) of a weight matrix WWW, which corresponds to the Lipschitz constant.

#### **Why It Helps:**

* Stabilizes the gradient flow during backpropagation by ensuring bounded activations.
* Prevents the discriminator from making overly sharp decisions, allowing the generator to learn effectively.
* Ensures that the GAN adheres to the Lipschitz continuity requirement of Wasserstein GAN (WGAN) formulations.

### **Class-Conditional Batch Normalization (CCBN)**

#### **Purpose:**

* Make Batch Normalization **conditional** on class labels, enabling the generator to produce class-specific features.
* Improve the generator’s ability to generate diverse images for different classes.

#### **How It Works:**

* In standard Batch Normalization, the scale (γ\gammaγ) and shift (β\betaβ) parameters are learned for the entire batch.
* In **Class-Conditional Batch Normalization**, the γ\gammaγ and β\betaβ parameters are conditioned on the class label.

Given a class label yyy, the formulation becomes:

y′=x−μσ⋅γ(y)+β(y)y' = \frac{x - \mu}{\sigma} \cdot \gamma(y) + \beta(y)y′=σx−μ​⋅γ(y)+β(y)

Where:

* γ(y)\gamma(y)γ(y) and β(y)\beta(y)β(y) are embeddings derived from the class label.
* μ\muμ: Mean of the batch.
* σ\sigmaσ: Standard deviation of the batch.

### **Truncation Trick**

#### **Purpose:**

* Trade off between sample **quality** and **diversity** in the generator's outputs.
* Useful when generating higher-quality images is prioritized over diversity.

#### **How It Works:**

* The **truncation trick** constrains the latent space sampling by truncating the latent vector zzz (sampled from a Gaussian distribution) to a smaller range.
* Instead of sampling z∼N(0,I)z \sim \mathcal{N}(0, I)z∼N(0,I), truncation modifies zzz as:

z′=clip(z,−truncation,truncation)z' = \text{clip}(z, -\text{truncation}, \text{truncation})z′=clip(z,−truncation,truncation)

Where **truncation** is a hyperparameter controlling the range of sampling.

#### **Effect:**

* **Lower truncation** values reduce the diversity of zzz, leading to higher-quality but less diverse samples.
* **Higher truncation** values allow for more diverse sampling but can result in lower-quality samples.

#### **Why It Helps:**

1. Reduces noise in the latent space, allowing the generator to focus on well-defined modes.
2. Provides control over the balance between diversity and quality in the generated samples.
3. Allows fine-tuning for specific use cases (e.g., high-quality image generation).