TensorFlow Model Garden aims to provide reliable state-of-the-art machine learning models to demonstrate best practices in TensorFlow 2 and illustrate end-to-end real-world use cases.

Your code should follow our guidelines to uphold our objectives of readable, usable, and maintainable code.

We provide the following principles and coding guidelines to ensure simple and readable code with high-quality models in the Model Garden.

What to do

- Use TensorFlow 2's high-level API standards to deliver reusable components.
  - Follow Keras Guiding principles and Guide to the Keras Functional API.
- Use `tf.keras` high-level APIs such as `tf.keras.Model` , `tf.keras.layers`
  - Check the guide for writing custom layers and models with Keras.
- Recommend to use Keras APIs such as `model.compile()` , `model.fit()` , `model.evaluate()` , `model.predict()` for model configuration, training, and evaluation loops.
  - If you need to write your own training & evaluation loops, see this example code.
  - However, more strict requirements including the TensorFlow code usability review will be enforced to have unified training loops.
- Use the TensorFlow distribution strategy API (`tf.distribute.Strategy`) to distribute training workloads to single-host/multi-accelerator as well as multi-host/multi-accelerator configurations.
  - See the distributed training guide.
- Provide full-coverage unit tests for your code.
- Provide modular and reusable code with the best effort.
- Conform to [[code style]] guide.
- Use Python 3

Note: Exceptions can be made case by case basis.

What NOT to do

- Any code changes that harm usability
- Any complicated custom training loop implementations

Data Dependencies

- The TensorFlow Model Garden is a code-only repository. Your data or image files should be hosted elsewhere (e.g., Google Cloud Storage bucket).

Code examples and templates

- Code examples and templates will be provided here soon.