Adversarial domain adaptation

- Two domains
  - e.g. mnist digits Vs actual digits on photos
- First domain is labeled, second isn't
- Wanna learn for the second domain

Adversarial domain adaptation

- Two domains
  - handwritten digits
  - house number digits
- First domain is labeled, second isn't
- You want your model to work on the second domain

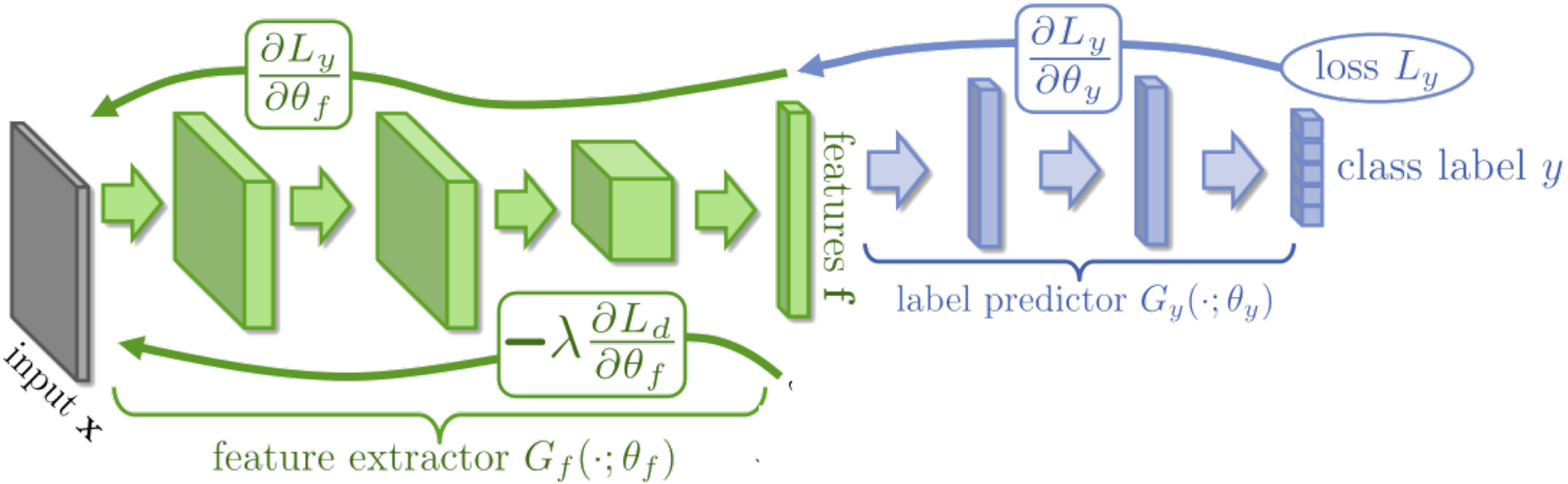
Labeled data

Unlabeled data

From the arxiv paper:
Top-performing deep architectures are trained on massive amounts of labeled data. In the absence of labeled data for a certain task, domain adaptation often provides an attractive option given that labeled data of similar nature but from a different domain (e.g. synthetic images) are available

Domain adaptation

Idea: discriminator should not be able to distinguish features on two domains



Source: https://arxiv.org/abs/1409.7495

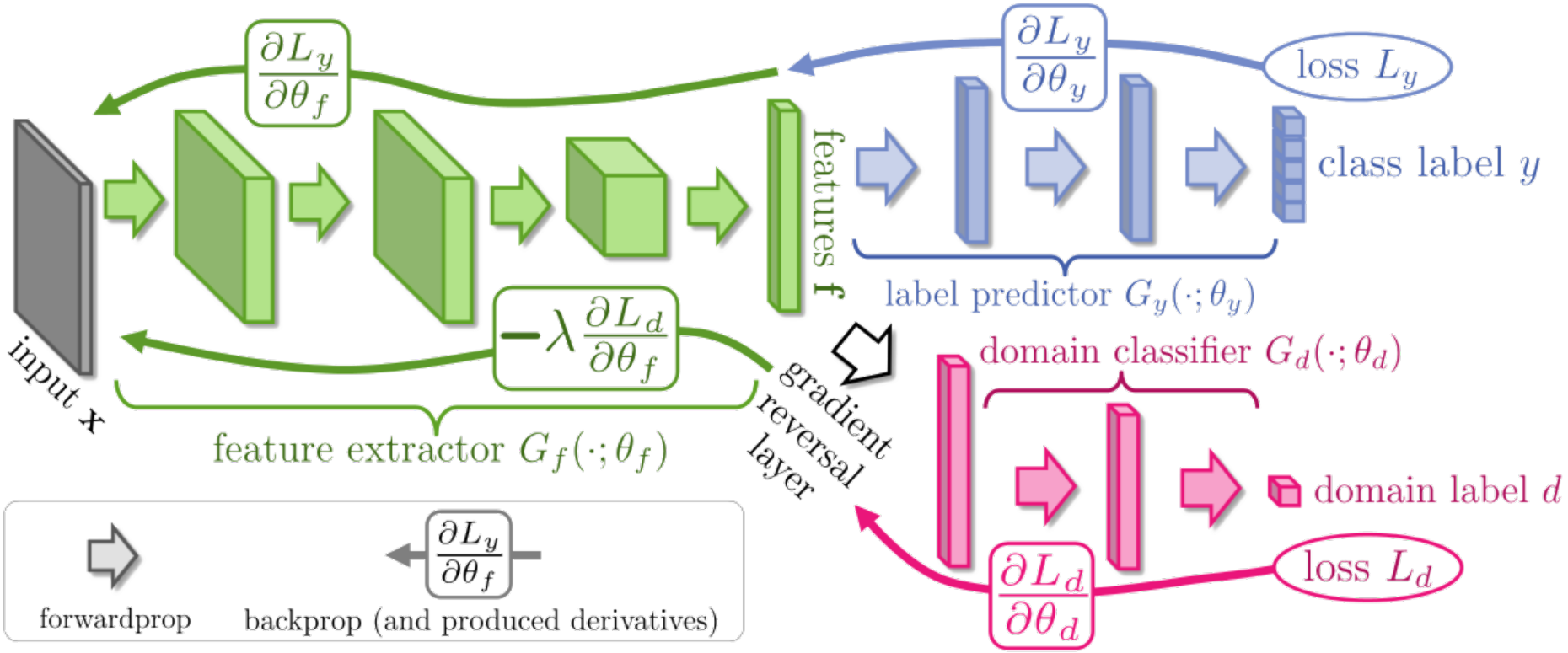
Domain adaptation

From arxiv paper:

While the parameters of the label predictor and domain classifier are optimized in order to minimize their error on the training set, the parameters of the underlying deep feature mapping are optimized in order to minimize the loss of the label classifier and to maximize the loss of the domain classifier. The latter encourages domain-invariant features to emerge in the course of the optimization.

DOMAIN INVARIANT FEATURES is key.

Idea: discriminator should not be able to distinguish features on two domains



Source: https://arxiv.org/abs/1409.7495

Domain adaptation

Idea: discriminator should not be able to distinguish features on two domains

Want discriminator to classify original (real) domain samples as real domain.

Want disc. to classify x_mc samples as from the 'fake' domain.

$$-logP\left(real|h(x_{real})\right) - log\left[1 - P\left(real|h(x_{mc})\right)\right] \rightarrow \min_{discriminator}$$

$$L_{classifier}\left(y_{mc}, y\left(h(x_{mc})\right)\right) - logP\left(real|h(x_{mc})\right) \rightarrow \min_{classifier}$$

standard classifier loss between target and predicted

As classifier, we want to fool discriminator into think that our mc samples are REAL.
This is what they mean by training the 'deep feature mapping' to be invariant to first and second domain.

y_{mc} is the label., x_{mc} is the input data.

L_{classifier} is the main classifier loss.

h(x_{mc}) is the output of the feature extractor.
y(h(x_{mc})) is the label of the x_{mc} sample.

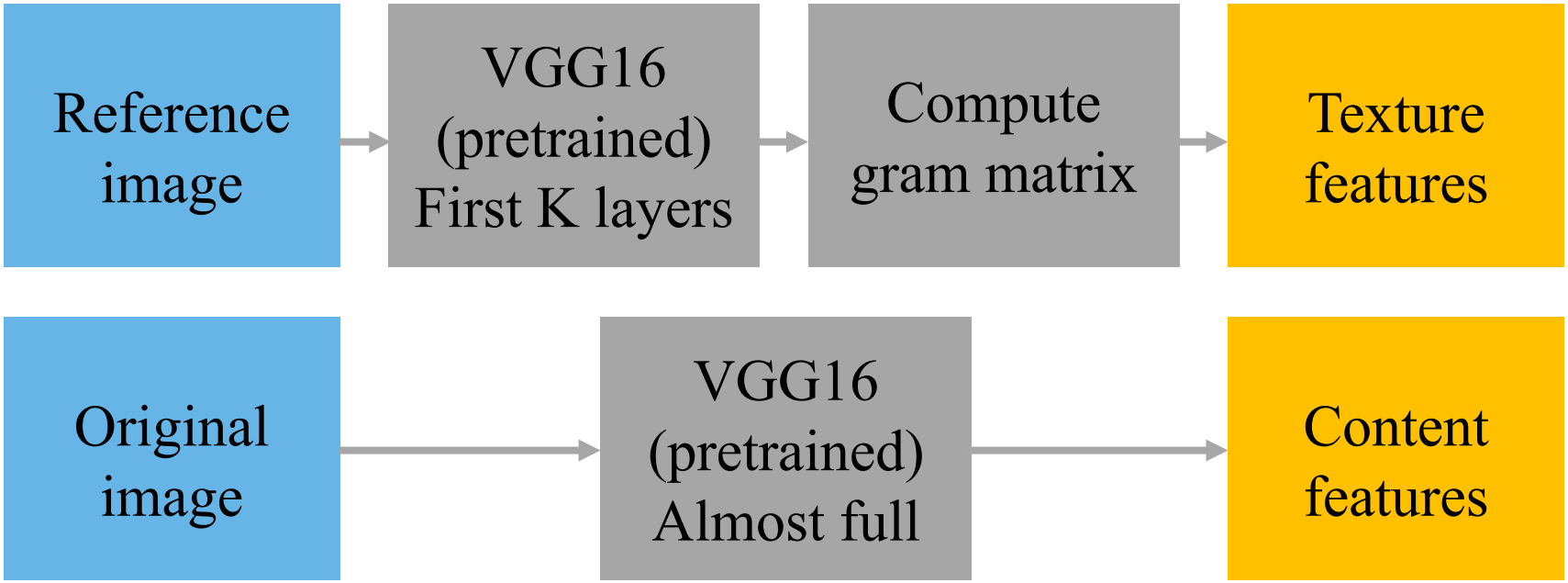
TODO: read tthe domain adaptation paper.

Art style transfer

We want to maintain the integrity of the image. We want the image drawn in a specific 'texture', but nevertheless have the same contents - not random.

Formulate and optimize texture loss

$$L = \|Texture(x_{ref}) - Texture(x_{cand})\| + \|Content(x_{orig}) - Content(x_{cand})\|$$



Art style transfer

+

=