Neural Inverse Knitting: From Images to Manufacturing Instructions

Alexandre Kaspar*, Tae-Hyun Oh*, Liane Makatura, Petr Kellnhofer and Wojciech Matusik
Industrial Knitting
Example:
Sweater
[Ministry of Supply]
Example: Scarf

[Kniterate]
Industrial Knitting

- Whole garments from scratch
Industrial Knitting

- Control of individual needles
- Whole garments from scratch
Knitted Garment & Patterns

Many garments are knitted:
• Beanies, scarves
• Gloves, socks and underwear
• Sweaters, sweatpants

Current machines can create those garments **seamlessly** (no sewing needed).
Knitted Garment & Patterns

Those garments have various types of surface patterns (knitting patterns).

These can be fully controlled by industrial knitting machine.

= User customization!
Machine Knitting Programming

Low-level machine code requires skilled experts

= Knitting masters
Scenario

1. User takes picture of knitting pattern
Scenario

1. User takes picture of knitting pattern
2. System creates knitting instructions
Scenario

1. User takes picture of knitting pattern
2. System creates knitting instructions
3. User reuses pattern for new garment
Machine Knitting

Brief background
Machine Knitting Terminology

(a) woven fabric
(b) weft knitted fabric
(c) warp knitted fabric

Illustration from [Underwood09]
Machine Knitting Terminology

Illustration from [Underwood09]
Machine Knitting Terminology

V-bed machine & knitting bed
Machine Knitting Terminology

- Yarn carrier
- Yarn
- Back needle bed
- Front needle bed
- Individual needles

Illustration from [Narayanan18]
Machine Knitting Operations

“Tuck” operation

Illustration from [Narayanan18]
Machine Knitting Operations

“Knit” operation

Illustration from [Narayanan18]
Machine Knitting Operations

“Transfer” operation

Illustration from [Narayanan18]
Machine Knitting Operations

Racking = offsetting between the two beds

Racking = 0

Racking ~ 6
The Data & its Acquisition

For 2D machine knitting programs
2D Knitting Pattern Programs

Image of 20x20 pattern

20x20 pattern program

“Pixels” are per-needle instructions over time
Knitting Pattern DSL

Domain Specific Language (DSL) for regular knitting patterns

- **Basic operations**: K, P, T, M
- **Cross operations**: XR+, XR-, XL+, XL-
- **Move operations**: FR1, FR2, FL1, FL2, BR1, BR2, BL1, BL2
- **Stack Order**: S
Knitting Pattern DSL

Domain Specific Language (DSL) for regular knitting patterns

Cross operations

XR+  XR-  XL+  XL-

Basic operations

Knit + …

Move operations

FR1  FR2  FL1  FL2  BR1  BR2  BL1  BL2

Stack Order

Ω  Ω
Full rows of operations are executed at once with the following sequence:

1. Move “current stitches” to the operation side (front | back)
2. Apply “needle operation” (knit | tuck | miss)
3. Transfer moving stitches to back bed (cross | move | stack)
4. Apply *sequence* of moves depending on the operations (cross | move)
5. Bring back all stitches to front bed (purl | cross | move | stack)
Full rows of operations are executed at once with the following sequence:

1. Move “current stitches” to the operation side (front | back)
2. Apply “needle operation” (knit | tuck | miss)
3. Transfer moving stitches to back bed (cross | move | stack)
4. Apply sequence of operation-related moves (cross | move)
5. Bring back all stitches to front bed (purl | cross | move | stack)

Encoded by operation type:
- Move = relative order not important
- Cross = relative order defined by group and “order” (upper | lower)
Dataset: Initial Attempt

Individual 20x20 patterns

High-quality registration
- From color frame
- Still not per-stitch…

Total: ~200 patterns
Time: ~1 month (intern)
= not enough data!
Dataset: Better Attempt

Capture setup with steel rods to normalize tension
Dataset Content

- Paired instructions with real (2,088) and synthetic (14,440) images.
- Synthetic data from automatic screen capture of KnitPaint (Shima’s software)
Machine Learning Details

Using two different types of supervision data
Learning Problem

Mapping **images** to discrete **instruction maps**

= CE loss minimization

Using two domains of data (one real, one synthetic)

= How to best combine both
Generalization Bound with Two Domains

With probability at least $1 - \delta$

\[
\frac{1}{2} \left| \mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h^*_T, y) \right| \leq \alpha \left( \text{disc}_H(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon
\]
Generalization Bound with Two Domains

With probability at least $1 - \delta$

$$\frac{1}{2} \left| \mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h^*_T, y) \right| \leq \alpha \left( \text{disc}_H(D_S, D_T) + \lambda \right) + \epsilon$$

Empirical min. $\arg \min_h \alpha \mathcal{L}_S(h, y) + (1 - \alpha) \mathcal{L}_T(h, y)$
Generalization Bound with Two Domains

With probability at least $1 - \delta$

$$\frac{1}{2} \left| \mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h_T^*, y) \right| \leq \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon$$
Generalization Bound with Two Domains

With probability at least $1 - \delta$

$$\frac{1}{2} \left| \mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h^*_T, y) \right|$$

$$\leq \alpha \left( \text{disc}_H(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon$$

$$\epsilon(m, \alpha, \beta, \delta) = \sqrt{\frac{1}{2m} \left( \frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta} \right)} \log \left( \frac{2}{\delta} \right),$$

Hyper-parameter dependent term
Generalization Bound with Two Domains

With probability at least $1 - \delta$

$$\frac{1}{2} | \mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h^*_T, y) |$$

$$\leq \alpha \left( \text{disc}_H(D_S, D_T) + \lambda \right) + \epsilon$$

$$\lambda = \min_{h \in \mathcal{H}} \mathcal{L}_S(h, y) + \mathcal{L}_T(h, y).$$

Ideal error of the combined losses
Generalization Bound with Two Domains

With probability at least $1 - \delta$

$$\frac{1}{2} \left| \mathcal{L}_T(h', y) - \mathcal{L}_T(h^*_T, y) \right| \leq \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon$$

Discrepancy between distributions

$$\text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) = \max_{h, h' \in \mathcal{H}} \left| \mathcal{L}_{\mathcal{D}_S}(h, h') - \mathcal{L}_{\mathcal{D}_T}(h, h') \right|$$
Data distributions

• Two different distribution types

$D_S$ Real data

$D_T$ Synthetic data
Data distributions

- Two different distribution types

\[ D_S \] Real data

\[ D_T \] Synthetic data
From synthetic to real

\[ \min_M \text{disc}(D_S, M(D_T)) \]

- S+U Learning [Shrivastava’17]

Real data

Synthetic data

\[ M(\cdot) \]
From synthetic to real

- S+U Learning [Shrivastava’17]

\[
\min_{M} \text{disc}(\mathcal{D}_S, M(\mathcal{D}_T))
\]

Real-looking data

Synthetic data
From synthetic to real

• One-to-many mapping!

$\min_M \text{disc}(D_S, M(D_T))$

$M(\cdot)$
From synthetic to real

• One-to-many!

\[
\min_{M} \text{disc}(D_S, M(D_T))
\]

Tension  Lighting  Color  Yarn

D
From real to synthetic

• Many-to-one!

\[ \min_M \text{disc}(M(D_S), D_T) \]

\[ M(\cdot) \]

Tension  Lighting  Color  Yarn

Regular / Normalized
Network composition

Diagram showing the flow of data from Real to Refiner, then to Regularized, and finally to Img2Prog, leading to a Program.
Results

Qualitative and quantitative evaluation
Architecture variations

Input

Ground Truth

(b1) Img2prog with CE

(b2) Img2prog with MILCE

(d1) Refiner + Img2prog++
## Baseline, comparisons and impact of mixing

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Perceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full</td>
<td>FG</td>
</tr>
<tr>
<td>(a1) CycleGAN (Zhu et al., 2017)</td>
<td>57.27</td>
<td>24.10</td>
</tr>
<tr>
<td>(a2) Pix2Pix (Isola et al., 2017)</td>
<td>56.20</td>
<td>47.98</td>
</tr>
<tr>
<td>(a3) UNet (Ronneberger et al., 2015)</td>
<td>89.65</td>
<td>63.99</td>
</tr>
<tr>
<td>(a4) Scene Parsing (Zhou et al., 2018)</td>
<td>91.58</td>
<td>73.95</td>
</tr>
<tr>
<td>(a5) S+U (Shrivastava et al., 2017)</td>
<td>91.32</td>
<td>71.00</td>
</tr>
<tr>
<td>(b1) Img2prog (real only) with CE</td>
<td>91.57</td>
<td>71.37</td>
</tr>
<tr>
<td>(b2) Img2prog (real only) with MILCE</td>
<td>91.74</td>
<td>72.30</td>
</tr>
<tr>
<td>(c1) Refiner + Img2prog (α = 0.9)</td>
<td>93.48</td>
<td>78.53</td>
</tr>
<tr>
<td>(c2) Refiner + Img2prog (α = 2/3)</td>
<td>93.58</td>
<td>78.57</td>
</tr>
<tr>
<td>(c3) Refiner + Img2prog (α = 0.5)</td>
<td>93.57</td>
<td>78.30</td>
</tr>
<tr>
<td>(c4) Refiner + Img2prog (α = 1/3)</td>
<td>93.19</td>
<td>77.80</td>
</tr>
<tr>
<td>(c5) Refiner + Img2prog (α = 0.1)</td>
<td>92.42</td>
<td>74.15</td>
</tr>
<tr>
<td>(d1) Refiner + Img2prog++ (α = 0.5)</td>
<td>94.01</td>
<td>80.30</td>
</tr>
</tbody>
</table>
Baseline, comparisons and impact of mixing

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Perceptual</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full</td>
<td>FG</td>
<td>SSIM</td>
<td>PSNR [dB]</td>
</tr>
<tr>
<td>(a1) CycleGAN (Zhu et al., 2017)</td>
<td>57.27</td>
<td>24.10</td>
<td>0.670</td>
<td>15.87</td>
</tr>
<tr>
<td>(a2) Pix2Pix (Isola et al., 2017)</td>
<td>56.20</td>
<td>47.98</td>
<td>0.660</td>
<td>15.95</td>
</tr>
<tr>
<td>(a3) UNet (Ronneberger et al., 2015)</td>
<td>89.65</td>
<td>63.99</td>
<td>0.847</td>
<td>21.21</td>
</tr>
<tr>
<td>(a4) Scene Parsing (Zhou et al., 2018)</td>
<td>91.58</td>
<td>73.95</td>
<td>0.876</td>
<td>22.64</td>
</tr>
<tr>
<td>(a5) S+U (Shrivastava et al., 2017)</td>
<td>91.32</td>
<td>71.00</td>
<td>0.864</td>
<td>21.42</td>
</tr>
<tr>
<td>(b1) Img2prog (real only) with CE</td>
<td>91.57</td>
<td>71.37</td>
<td>0.866</td>
<td>21.62</td>
</tr>
<tr>
<td>(b2) Img2prog (real only) with MILCE</td>
<td>91.74</td>
<td>72.30</td>
<td>0.871</td>
<td>21.58</td>
</tr>
<tr>
<td>(c1) Refiner + Img2prog (α = 0.9)</td>
<td>93.48</td>
<td>78.53</td>
<td>0.894</td>
<td>23.28</td>
</tr>
<tr>
<td>(c2) Refiner + Img2prog (α = 2/3)</td>
<td>93.58</td>
<td>78.57</td>
<td>0.892</td>
<td>23.27</td>
</tr>
<tr>
<td>(c3) Refiner + Img2prog (α = 0.5)</td>
<td>93.57</td>
<td>78.30</td>
<td>0.895</td>
<td>23.24</td>
</tr>
<tr>
<td>(c4) Refiner + Img2prog (α = 1/3)</td>
<td>93.19</td>
<td>77.80</td>
<td>0.888</td>
<td>22.72</td>
</tr>
<tr>
<td>(c5) Refiner + Img2prog (α = 0.1)</td>
<td>92.42</td>
<td>74.15</td>
<td>0.881</td>
<td>22.27</td>
</tr>
<tr>
<td>(d1) Refiner + Img2prog++ (α = 0.5)</td>
<td>94.01</td>
<td>80.30</td>
<td>0.899</td>
<td>23.56</td>
</tr>
</tbody>
</table>
## Baseline, comparisons and impact of mixing

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Perceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full</td>
<td>FG</td>
</tr>
<tr>
<td>(a1) CycleGAN (Zhu et al., 2017)</td>
<td>57.27</td>
<td>24.10</td>
</tr>
<tr>
<td>(a2) Pix2Pix (Isola et al., 2017)</td>
<td>56.20</td>
<td>47.98</td>
</tr>
<tr>
<td>(a3) UNet (Ronneberger et al., 2015)</td>
<td>89.65</td>
<td>63.99</td>
</tr>
<tr>
<td>(a4) Scene Parsing (Zhou et al., 2018)</td>
<td>91.58</td>
<td>73.95</td>
</tr>
<tr>
<td>(a5) S+U (Shrivastava et al., 2017)</td>
<td>91.32</td>
<td>71.00</td>
</tr>
<tr>
<td>(b1) Img2prog (real only) with CE</td>
<td>91.57</td>
<td>71.37</td>
</tr>
<tr>
<td>(b2) Img2prog (real only) with MILCE</td>
<td>91.74</td>
<td>72.30</td>
</tr>
<tr>
<td>(c1) Refiner + Img2prog ($\alpha = 0.9$)</td>
<td>93.48</td>
<td>78.53</td>
</tr>
<tr>
<td>(c2) Refiner + Img2prog ($\alpha = 2/3$)</td>
<td>93.58</td>
<td>78.57</td>
</tr>
<tr>
<td>(c3) Refiner + Img2prog ($\alpha = 0.5$)</td>
<td>93.57</td>
<td>78.30</td>
</tr>
<tr>
<td>(c4) Refiner + Img2prog ($\alpha = 1/3$)</td>
<td>93.19</td>
<td>77.80</td>
</tr>
<tr>
<td>(c5) Refiner + Img2prog ($\alpha = 0.1$)</td>
<td>92.42</td>
<td>74.15</td>
</tr>
<tr>
<td>(d1) Refiner + Img2prog++ ($\alpha = 0.5$)</td>
<td>94.01</td>
<td>80.30</td>
</tr>
</tbody>
</table>
Baseline, comparisons and impact of mixing

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Perceptual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Full</td>
<td>FG</td>
</tr>
<tr>
<td>CycleGAN (Zhu et al., 2017)</td>
<td>57.27</td>
<td>24.10</td>
</tr>
<tr>
<td>Pix2Pix (Isola et al., 2017)</td>
<td>56.20</td>
<td>47.98</td>
</tr>
<tr>
<td>UNet (Ronneberger et al., 2015)</td>
<td>89.65</td>
<td>63.99</td>
</tr>
<tr>
<td>Scene Parsing (Zhou et al., 2018)</td>
<td>91.58</td>
<td>73.95</td>
</tr>
<tr>
<td>S+U (Shrivastava et al., 2017)</td>
<td>91.32</td>
<td>71.00</td>
</tr>
<tr>
<td>Img2prog (real only) with CE</td>
<td>91.57</td>
<td>71.37</td>
</tr>
<tr>
<td>Img2prog (real only) with MILCE</td>
<td>91.74</td>
<td>72.30</td>
</tr>
<tr>
<td>Refiner + Img2prog (α = 0.9)</td>
<td>93.48</td>
<td>78.53</td>
</tr>
<tr>
<td>Refiner + Img2prog (α = 2/3)</td>
<td>93.58</td>
<td>78.57</td>
</tr>
<tr>
<td>Refiner + Img2prog (α = 0.5)</td>
<td>93.57</td>
<td>78.30</td>
</tr>
<tr>
<td>Refiner + Img2prog (α = 1/3)</td>
<td>93.19</td>
<td>77.80</td>
</tr>
<tr>
<td>Refiner + Img2prog (α = 0.1)</td>
<td>92.42</td>
<td>74.15</td>
</tr>
<tr>
<td>Refiner + Img2prog++ (α = 0.5)</td>
<td>94.01</td>
<td>80.30</td>
</tr>
</tbody>
</table>
How much data is enough data?

- Full Accuracy (%)
  - 200 samples (12.5%): 86.36
  - 400 samples (25%): 88.02
  - 800 samples (50%): 90.01
  - All samples (100%): 91.57

- Foreground Accuracy (%)
  - 200 samples (12.5%): 49.72
  - 400 samples (25%): 56.31
  - 800 samples (50%): 65.91
  - All samples (100%): 71.37
Limitations

And potential solutions
Issue of scale, stretch and orientation

We assume a specific scale, stretch (of 20x20 stitches) and a bottom-up orientation of stitch courses.

**Options:**

- Explicit model scale, stretch and orientation
  = makes training more complicated

- Separate selection (using measure of “confidence”)
  = take large-scale image, and try space of scales / stretches / rot.
Attempt at scale selection (successful)
Input variety

We only used Tamm 2/30 acrylic yarn.
How do we scale to more data, and more varieties of it?

Options:

○ **Simulation**: need fast simulation of yarn (hard, or slow), hopefully as a differentiable renderer (within the network)

○ **Online yarn images**: unsupervised way? Cycle-consistency? Additional side/weaker/stronger task?
Modeling Hard Constraints

Currently, output may have invalid instruction combinations. Tried to use penalty on valid 1st order neighborhood, but little impact.

Questions:
- How do we model hard constraints with a neural network?
- Split translation into instruction “potentials” and then select the actual instructions (e.g., using known knittability constraints)?
- Can we infer the syntax constraints automatically?
  - Note: non-trivial to specify beyond first-order neighborhood unless enough data is available…
Result Details

The great, the good, the not so good, and the ugly
Details: Perfect cases

Program

Knitted Sample
Details: Minor errors (no semantic issue)
Details: Larger errors (but knittable)
Details: Larger errors (but knittable)
Details: (few) catastrophic failures (only 2)

Not knittable!

Yarn collapses!
Past and Future Work

Where it came from, and where it is going
Past: Foundry - Multi-Material 3D Printing
Recent: Knitting Skeletons - CAD for Knitting
Now: Knit Sketching - Sketches within CAD

Work with sketches
- Wale flow
- Connectivity
- Stitch density
- Layers (sketches)
- Layers (patterns)

Generate data for the CAD system.
(with some efficient parameterization)
Next: InverseKnit++

Use sketch input capability to learn to map full knitted “shapes” directly into low-level knitting programs.

- More instruction irregularities
- Issue of occlusion (two-sided shapes)
- Ambiguity between shape and patterns
http://deepknitting.csail.mit.edu

Thank you for listening!
Dataset Details

Instruction distribution and accuracies
Dataset: instruction statistics

Figure 5. Instruction counts in descending order, for synthetic and real images. Note the logarithmic scale of the Y axis.
Table 2. Performance of Refined+Img2prog++ measured per instruction over the test set. This shows that even though our instruction distribution has very large variations, our network is still capable of learning some representation for the least frequent instructions (3 orders of magnitude difference for FR2, FL2, BR2, BL2 compared to K and P).

<table>
<thead>
<tr>
<th>Instruction</th>
<th>K</th>
<th>P</th>
<th>T</th>
<th>M</th>
<th>FR1</th>
<th>FR2</th>
<th>FL1</th>
<th>FL2</th>
<th>BR1</th>
<th>BR2</th>
<th>BL1</th>
<th>BL2</th>
<th>XR+</th>
<th>XR-</th>
<th>XL+</th>
<th>XL-</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy [%]</td>
<td>96.52</td>
<td>96.64</td>
<td>74.63</td>
<td>66.65</td>
<td>77.16</td>
<td>100.00</td>
<td>74.20</td>
<td>83.33</td>
<td>68.73</td>
<td>27.27</td>
<td>69.94</td>
<td>22.73</td>
<td>60.15</td>
<td>62.33</td>
<td>60.81</td>
<td>62.11</td>
<td>25.85</td>
</tr>
<tr>
<td>Frequency [%]</td>
<td>44.39</td>
<td>47.72</td>
<td>0.41</td>
<td>1.49</td>
<td>1.16</td>
<td>0.01</td>
<td>1.23</td>
<td>0.01</td>
<td>1.22</td>
<td>0.02</td>
<td>1.40</td>
<td>0.02</td>
<td>0.22</td>
<td>0.18</td>
<td>0.19</td>
<td>0.22</td>
<td>0.12</td>
</tr>
</tbody>
</table>
Architecture Details

Neural Networks and Losses
Actual Loss Function

2D cross entropy

Style loss + GAN loss
Actual Loss Function

Our combined loss is the weighted sum

\[ \mathcal{L} = \lambda_{\text{CE}} \mathcal{L}_{\text{CE}} + \lambda_{\text{Perc}} \mathcal{L}_{\text{Perc}} + \lambda_{\text{GAN}} \mathcal{L}_{\text{GAN}} \]  

(5)

where we used the weights: \( \lambda_{\text{CE}} = 3 \), \( \lambda_{\text{Perc}} = 0.02/(128)^2 \) and \( \lambda_{\text{GAN}} = 0.2 \). The losses \( \mathcal{L}_{\text{Perc}} \) and \( \lambda_{\text{GAN}} \) are measured on the output of Refiner, while the loss \( \lambda_{\text{CE}} \) is measured on Img2prog.
Figure 10. The illustration of the Refiner network architecture, where $S\#N$ denotes the stride size of $\#N$, IN_ReLU indicates the Instance normalization followed by ReLU, Resblk is the residual block that consists of ConvS1-ReLU-ConvS1 with shortcut connection (He et al., 2016), Upsample is the nearest neighbor upsampling with the factor 2$x$, $F$ is the output channel dimension. If not mentioned, the default parameters for all the convolutions are the stride size of 2, $F = 64$, and the $3 \times 3$ kernel size.
Theorem 1

About the Generalization Gap
Definition 1 (Discrepancy (Mansour et al., 2009)). Let $\mathcal{H}$ be a class of functions mapping from $\mathcal{X}$ to $\mathcal{Y}$. The discrepancy between two distribution $\mathcal{D}_1$ and $\mathcal{D}_2$ over $\mathcal{X}$ is defined as

$$\text{disc}_{\mathcal{H}}(\mathcal{D}_1, \mathcal{D}_2) = \max_{h, h' \in \mathcal{H}} |\mathcal{L}_{\mathcal{D}_1}(h, h') - \mathcal{L}_{\mathcal{D}_2}(h, h')|.$$  (6)

The discrepancy is symmetric and satisfies the triangle inequality, regardless of any loss function. This can be used to compare distributions for general tasks even including regression.
Lemma 1. Let $h$ be a hypothesis in class $\mathcal{H}$, and assume that $\mathcal{L}$ is symmetric and obeys the triangle inequality. Then

$$|\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)| \leq \alpha (\text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda),$$  \hspace{1cm} (7)

where $\lambda = \mathcal{L}_S(h^*, y) + \mathcal{L}_T(h^*, y)$, and the ideal joint hypothesis $h^*$ is defined as $h^* = \arg\min_{h \in \mathcal{H}} \mathcal{L}_S(h, y) + \mathcal{L}_T(h, y)$. 
Lemma 1: from Lemma 4 of [Ben-David 10]

\[\begin{align*}
|L_\alpha(h, y) - L_T(h, y)| &= \alpha |L_S(h, y) - L_T(h, y)| \\
&= \alpha |L_S(h, y) - L_S(h^*, h) + L_S(h^*, h) \\
&\quad - L_T(h^*, h) + L_T(h^*, h) - L_T(h, y)| \\
&\leq \alpha |L_S(h, y) - L_S(h^*, h)| + \\
&\quad |L_S(h^*, h) - L_T(h^*, h)| + |L_T(h^*, h) - L_T(h, y)| \\
&\leq \alpha |L_S(h^*, y) + L_S(h^*, h) - L_T(h^*, h)| + L_T(h^*, y) \\
&\leq \alpha (\text{disc}_H(\mathcal{D}_S, \mathcal{D}_T) + \lambda). \quad (8)
\end{align*}\]
Lemma 1: from Lemma 4 of [Ben-David 10]

Proof. The proof is based on the triangle inequality of $\mathcal{L}$, and the last inequality follows the definition of the discrepancy.

$$|\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)|$$

$$= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_T(h, y)|$$

$$= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h) + \mathcal{L}_S(h^*, h)$$

$$- \mathcal{L}_T(h^*, h) + \mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)|$$

$$\leq \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h)| +$$

$$|\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)|$$

$$\leq \alpha |\mathcal{L}_S(h^*, y) + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + \mathcal{L}_T(h^*, y)|$$

$$\leq \alpha (\text{disc}_H(\mathcal{D}_S, \mathcal{D}_T) + \lambda). \quad (8)$$
Lemma 1: from Lemma 4 of [Ben-David 10]

Proof. The proof is based on the triangle inequality of $\mathcal{L}$, and the last inequality follows the definition of the discrepancy.

\[
|\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)| \\
= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_T(h, y)| \\
= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h) + \mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \\
\leq \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h)| + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \\
\leq \alpha |\mathcal{L}_S(h^*, y) + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, y)| \\
\leq \alpha (\text{disc}_H(D_S, D_T) + \lambda).
\] (8)
Lemma 1: from Lemma 4 of [Ben-David 10]

Proof. The proof is based on the triangle inequality of $\mathcal{L}$, and the last inequality follows the definition of the discrepancy.

\[
|\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)| \\
= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_T(h, y)| \\
= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h) + \mathcal{L}_S(h^*, h) \\
- \mathcal{L}_T(h^*, h) + \mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \\
\leq \alpha \left( |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h)| + \\
|\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \right) \\
\leq \alpha \left( |\mathcal{L}_S(h^*, y)| + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, y)\right) \\
\leq \alpha (\text{disc}_\mathcal{H}(D_S, D_T) + \lambda).
\]
Lemma 1: from Lemma 4 of [Ben-David 10]

Proof. The proof is based on the triangle inequality of $\mathcal{L}$, and the last inequality follows the definition of the discrepancy.

\[
\begin{align*}
|\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)| &= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_T(h, y)| \\
&= \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h) + \mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h) + \mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \\
&\leq \alpha |\mathcal{L}_S(h, y) - \mathcal{L}_S(h^*, h)| + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + |\mathcal{L}_T(h^*, h) - \mathcal{L}_T(h, y)| \\
&\leq \alpha |\mathcal{L}_S(h^*, y)| + |\mathcal{L}_S(h^*, h) - \mathcal{L}_T(h^*, h)| + \mathcal{L}_T(h^*, y) \\
&\leq \alpha (\text{disc}_H(\mathcal{D}_S, \mathcal{D}_T) + \lambda).
\end{align*}
\]
Lemma 2: from [Ben-David 10]

**Lemma 2** ((Ben-David et al., 2010)). For a fixed hypothesis \( h \), if a random labeled sample of size \( m \) is generated by drawing \( \beta m \) points from \( \mathcal{D}_S \) and \( (1-\beta)m \) points from \( \mathcal{D}_T \), and labeling them according to \( y_S \) and \( y_T \) respectively, then for any \( \delta \in (0, 1) \), with probability at least \( 1-\delta \) (over the choice of the samples),

\[
|\hat{\mathcal{L}}_\alpha(h, y) - \mathcal{L}_\alpha(h, y)| \leq \epsilon(m, \alpha, \beta, \delta),
\]

where \( \epsilon(m, \alpha, \beta, \delta) = \sqrt{\frac{1}{2m} \left( \frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta} \right) \log\left(\frac{2}{\delta}\right)} \).

The detail function form of \( \epsilon \) will be omitted for simplicity. We can fix \( m, \alpha, \beta, \) and \( \delta \) when the learning task is specified, then we can treat \( \epsilon(\cdot) \) as a constant.
Theorem 1: Generalization Gap

**Theorem 1.** Let $\mathcal{H}$ be a hypothesis class, and $S$ be a labeled sample of size $m$ generated by drawing $\beta m$ samples from $\mathcal{D}_S$ and $(1 - \beta)m$ samples from $\mathcal{D}_T$ and labeling them according to the true label $y$. Suppose $\mathcal{L}$ is symmetric and obeys the triangle inequality. Let $\hat{h} \in \mathcal{H}$ be the empirical minimizer of $\hat{h} = \arg\min_h \hat{\mathcal{L}}_\alpha(h, y)$ on $S$ for a fixed $\alpha \in [0, 1]$, and $h^*_T = \arg\min_h \mathcal{L}_T(h, y)$ the target error minimizer. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of the samples), we have

$$\frac{1}{2} |\mathcal{L}_T(\hat{h}, y) - \mathcal{L}_T(h^*_T, y)| \leq \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon,$$

where $\epsilon(m, \alpha, \beta, \delta) = \sqrt{\frac{1}{2m} \left( \frac{\alpha^2}{\beta} + \frac{(1 - \alpha)^2}{1 - \beta} \right) \log(\frac{2}{\delta})}$, and $\lambda = \min_{h \in \mathcal{H}} \mathcal{L}_S(h, y) + \mathcal{L}_T(h, y)$. 


Proof of Theorem 1

Proof. We use Lemmas 1 and 2 for the bound derivation with their associated assumptions.

\[ \mathcal{L}_T(\hat{h}, y) \]
\[ \leq \mathcal{L}_\alpha(\hat{h}, y) + \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right), \]  \hspace{1cm} (By Lemma 1) \hspace{1cm} (11)

\[ \leq \hat{\mathcal{L}}_\alpha(\hat{h}, y) + \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon, \]  \hspace{1cm} (By Lemma 2) \hspace{1cm} (12)

\[ \leq \hat{\mathcal{L}}_\alpha(h^*_T, y) + \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + \epsilon, \]  \hspace{1cm} (\hat{h} = \arg \min_{\hat{h} \in \mathcal{H}} \hat{\mathcal{L}}_\alpha(\hat{h})) \hspace{1cm} (13)

\[ \leq \mathcal{L}_\alpha(h^*_T, y) + \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + 2\epsilon, \]  \hspace{1cm} (By Lemma 2) \hspace{1cm} (14)

\[ \leq \mathcal{L}_T(h^*_T, y) + 2\alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) + 2\epsilon, \]  \hspace{1cm} (By Lemma 1) \hspace{1cm} (15)

Cost of “swapping” is at least ...

Specific sampling:
\[ |\mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y)| \leq \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) \]

Because \( \hat{\mathcal{L}}_\alpha(h, y) \leq \hat{\mathcal{L}}_\alpha(h^*_T, y) \)

Specific “de-sampling”:
\[ |\hat{\mathcal{L}}_\alpha(h, y) - \mathcal{L}_\alpha(h, y)| \leq \epsilon(m, \alpha, \beta, \delta) \]

\[ \mathcal{L}_\alpha(h, y) - \mathcal{L}_T(h, y) \leq \alpha \left( \text{disc}_\mathcal{H}(\mathcal{D}_S, \mathcal{D}_T) + \lambda \right) \]