



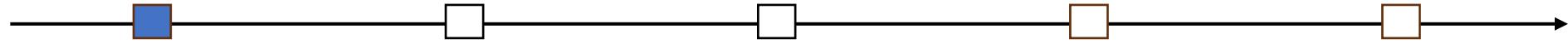
Why Do Deep Learning Projects Differ in Compatible Framework Versions? An Exploratory Study

Huashan Lei¹, Shuai Zhang¹, Jun Wang¹, Guanping Xiao¹
Yepang Liu², Yulei Sui³

¹ **Nanjing University of Aeronautics and Astronautics, China**

² Southern University of Science and Technology, China

³ University of New South Wales, Australia



Motivation

Methodology

Results

Implications

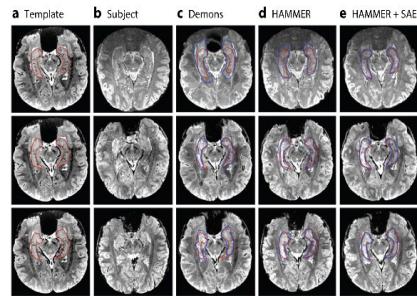
Conclusions



Deep learning (DL) is widely used in various domains



Self-driving Cars



A Shen D, et al. 2017.
B Annu. Rev. Biomed. Eng. 19:221–48

Medical Diagnosis



Natural Language Processing



Speech Recognition



Motivation

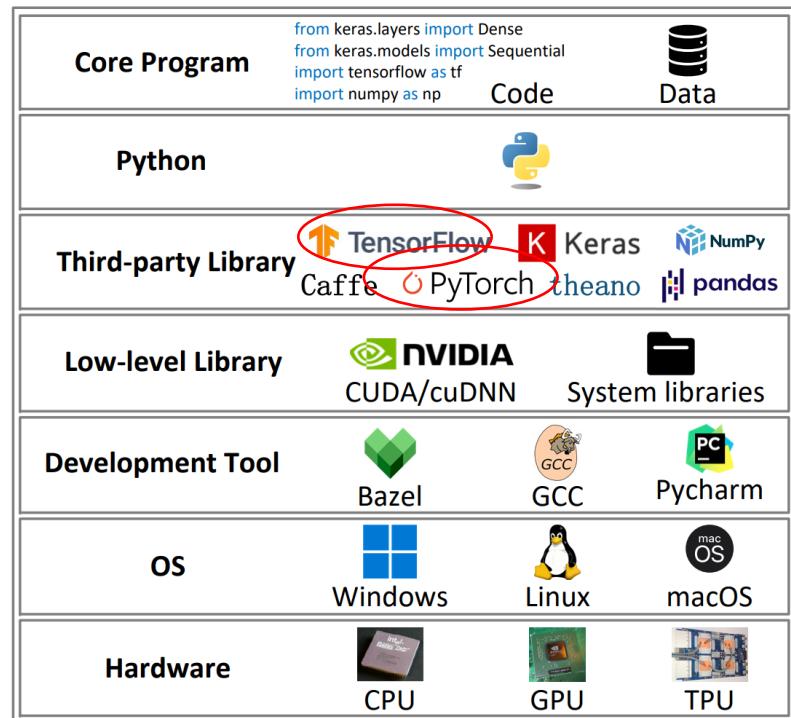
Methodology

Results

Implications

Conclusions

DL projects rely heavily on DL frameworks, which evolve frequently

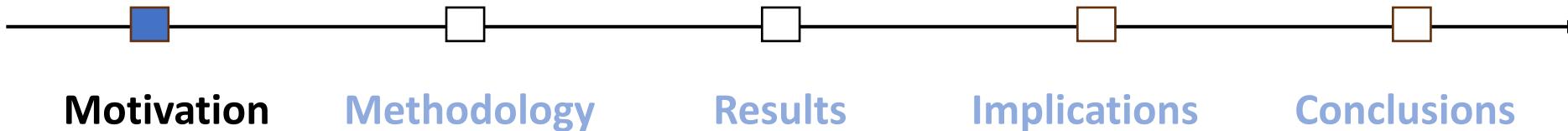


Components of DL Systems

13 TensorFlow

New releases per year (as of December 2022)

5 PyTorch



Prevalent compatibility issues in DL projects

from torch._C import * ImportError: DLL load failed:
The specified module could not be found

Asked 5 years, 4 months ago Modified 6 months ago Viewed 17k times

TypeError: __init__() got an unexpected keyword argument 'shape'

11 Asked 6 years, 3 months ago Modified 2 years, 6 months ago Viewed 14k times

TypeError: tensor() got an unexpected keyword argument 'names'

1 Asked 2 years, 3 months ago Modified 2 years, 3 months ago Viewed 2k times

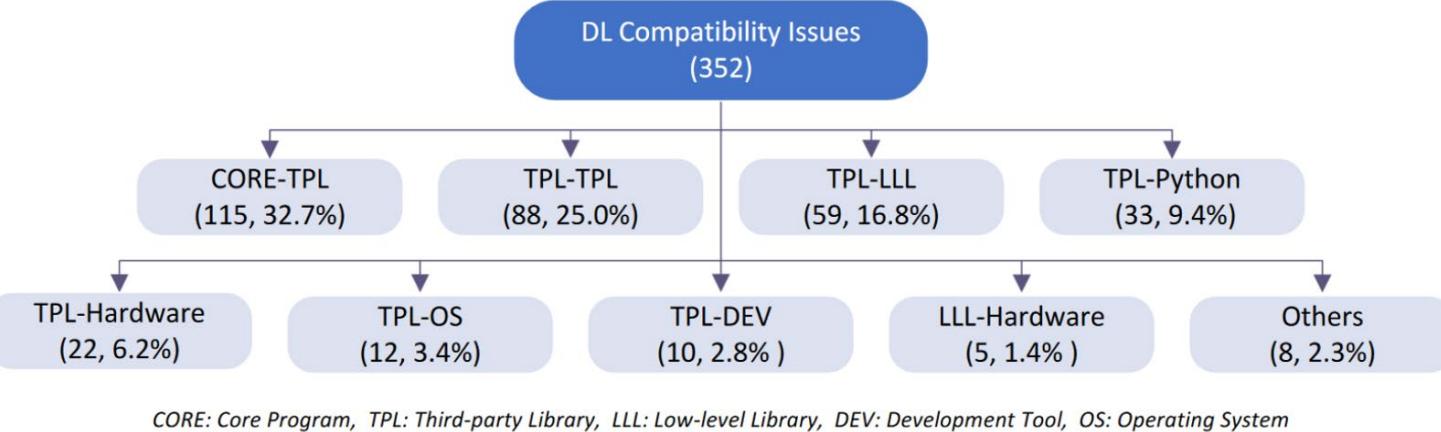
Tensorflow 2.0 - AttributeError: module 'tensorflow' has no attribute 'Session'

1 Asked 4 years, 7 months ago Modified 6 months ago Viewed 470k times

When I am executing the command `sess = tf.Session()` in Tensorflow 2.0 environment, I am getting an error message as below:

224

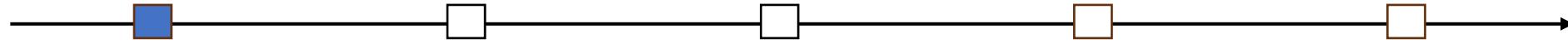
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
AttributeError: module 'tensorflow' has no attribute 'Session'



Type of Compatibility Issues in DL Systems

DL Compatibility Issue: incompatible interaction problem between components

Jun Wang, Guanping Xiao, Shuai Zhang, Huashan Lei, Yepang Liu and Yulei Sui. Compatibility Issues in Deep Learning Systems: Problems and Opportunities. ESEC/FSE 2023



Motivation

Methodology

Results

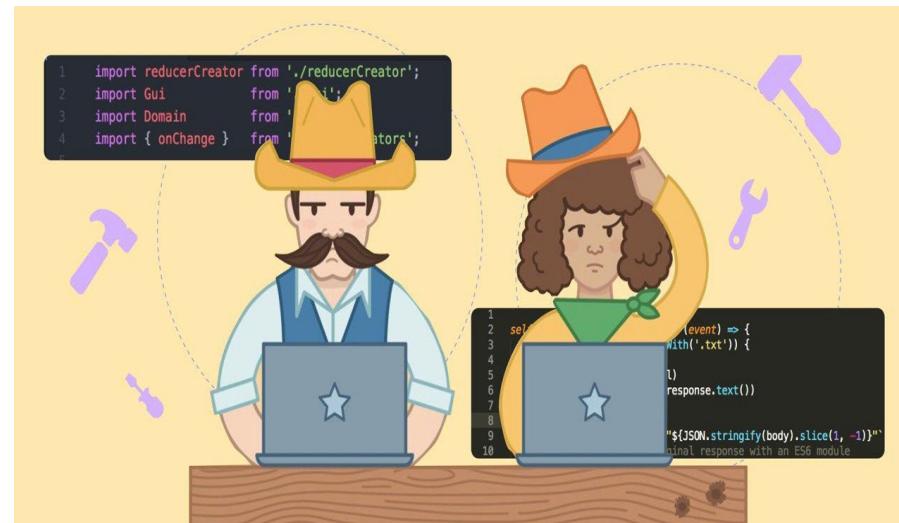
Implications

Conclusions

Difference in framework version compatibility (DFVC) among DL projects

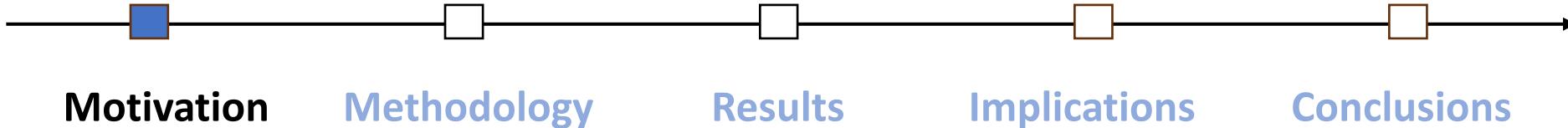


Execute normally



Crash

Differences in the programming experiences and the configuration of runtime environments



Use the same DL framework but have DFVC

- DL framework: **TensorFlow**
- Version: **2.0.0**
- Related API: **tf.Session()**

```
hwalsuklee/tensorflow-fast-style-transfer:  
1 import tensorflow as tf  
...  
197 sess = tf.Session(config=tf.ConfigProto(allow_soft_placement=True))  
...  
  
Traceback (most recent call last):  
File "run_train.py", line 224, in <module>  
    main()  
File "run_train.py", line 197, in main  
    sess = tf.Session(config=tf.ConfigProto(allow_soft_placement=True))  
AttributeError: module 'tensorflow' has no attribute 'Session'
```

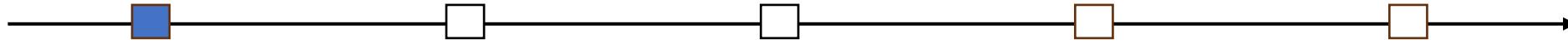
Crash

```
githubharald/SimpleHTR: model.py  
...  
6 import tensorflow as tf  
...  
153 sess = tf.compat.v1.Session() # TF session  
...  
  
...  
Init with stored values from ./model/snapshot-1  
Recognized: "word"  
Probability: 0.9937934279441833
```

Normal

(a) tensorflow-fast-style-transfer

(b) SimpleHTR



Motivation

Methodology

Results

Implications

Conclusions

Use the same DL framework but have DFVC

- DL framework: **TensorFlow**
- Version: **1.13.2**
- Related API: **tf.nn.conv2d()**

githubbarald/SimpleHTR: model.py

```
...
6   import tensorflow as tf
...
72  conv = tf.nn.conv2d(input=pool, filters=kernel, padding='SAME', strides=(1, 1, 1, 1))
...
```

Traceback (most recent call last):
File "main.py", line 202, in <module>
 main()
File "main.py", line 197, in main
 model = Model(char_list_from_file(), decoder_type, must_restore=True, dump=args.dump)
File "/home/.../SimpleHTR/src/model.py", line 43, in __init__
 self.setup_cnn()
File "/home/.../SimpleHTR/src/model.py", line 72, in setup_cnn
 conv = tf.nn.conv2d(input=pool, filters=kernel, padding='SAME', strides=(1, 1, 1, 1))
TypeError: conv2d() got an unexpected keyword argument 'filters'

Crash

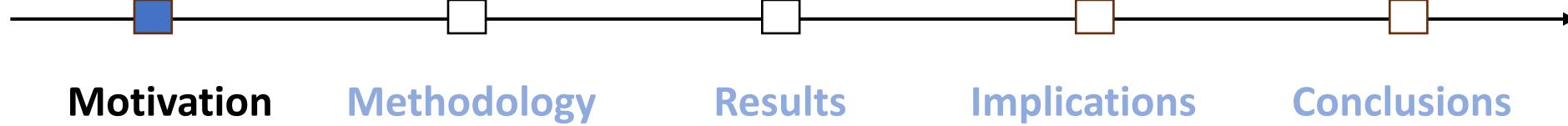
anishathalye/neural-style: vgg.py

```
...
5   import tensorflow.compat.v1 as tf
...
97 conv = tf.nn.conv2d(input, tf.constant(weights), strides=(1, 1, 1, 1), padding="SAME")
...
...
Iteration 1/ 1
content loss: 2.0134e+06
style loss: 6.78778e+07
tv loss: 23257.1
total loss: 6.99145e+07
```

Normal

(a) SimpleHTR

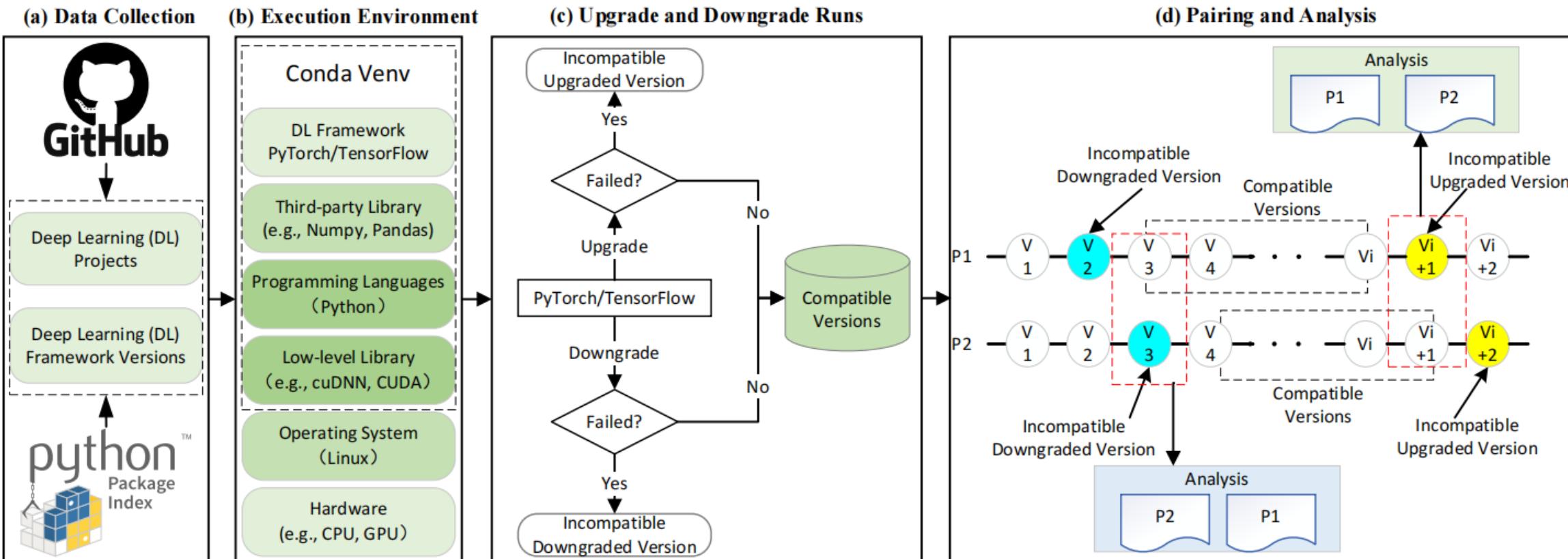
(b) neural-style



Research Questions

- RQ1:
How prevalent is it for DL projects to exhibit differences in compatible framework versions?
- RQ2:
What are the root causes for the difference in framework version compatibility among DL projects?

Overview of our empirical study





Motivation

Methodology

Results

Implications

Conclusions



Data Collection and Aggregation



- DL projects: Collected from GitHub.
- DL framework versions: Collected from PyPI repository.



Framework	Projects	Versions	Range	Time Frame
TensorFlow	50	66	0.12.0-2.8.0	03/31/22
PyTorch	90	20	1.0.0-1.11.0	03/31/22



Project Execution Environment Configuration

DL Framework
PyTorch/TensorFlow

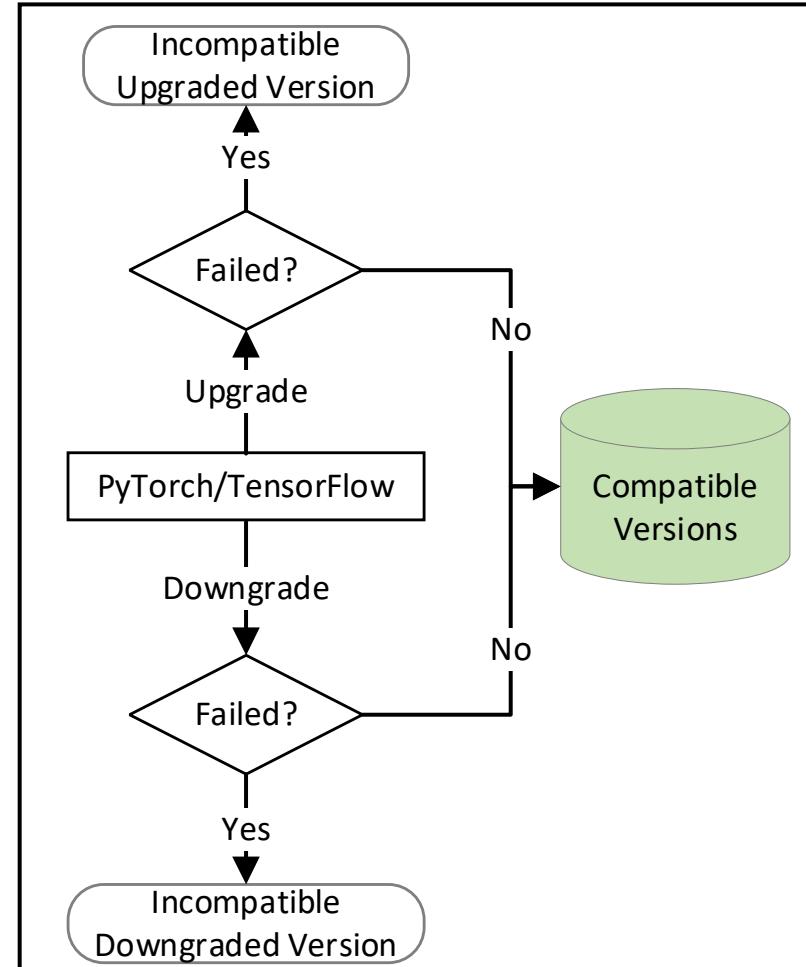
Third-party Library
(e.g., Numpy, Pandas)

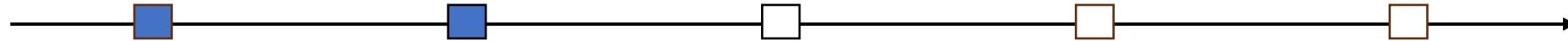
Programming Languages
(Python)

Low-level Library
(e.g., cuDNN, CUDA)

Upgrade and Downgrade Runs

- **Incompatible Upgraded Version (IUU):** We refer to **the initial framework version among two consecutive updates, where the project did not execute properly**, as the incompatible upgraded version (IUU).
- **Incompatible Downgraded Version (IDV):** We refer to **the initial framework version among two consecutive downgrades, where the project did not operate as intended**, as the incompatible downgraded version (IDV).
- **Compatible Versions:** The framework versions in which projects can run smoothly are considered compatible versions.





Motivation

Methodology

Results

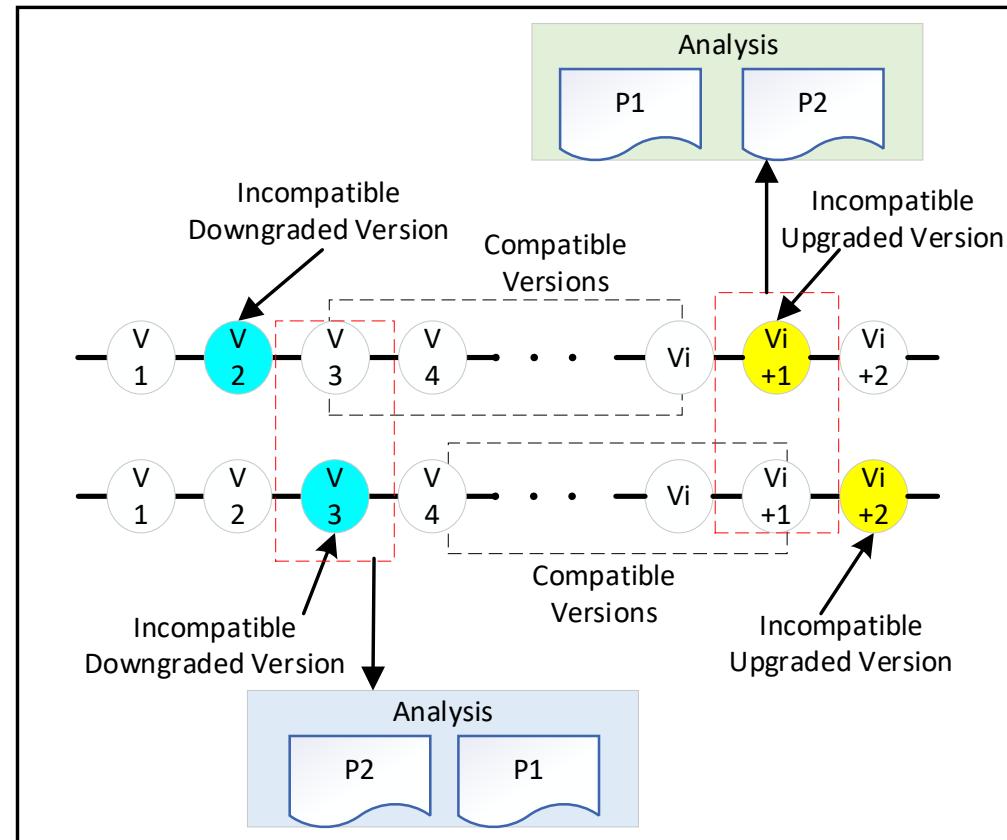
Implications

Conclusions

Pairing and Analysis

6,926 pairs and further classified them into three categories.

- (1) Project y and project x use different Python versions
- (2) Project y does not call the framework API involved in project x's error message
- (3) Project y calls the framework API involved in project x's error message



Category	(1) Python	(2) w/o. the same API	(3) w. the same API
#Pairs	1,415	4,627	884
%Percentage	20.4%	66.8%	12.8%



Motivation

Methodology

Results

Implications

Conclusions



Pairing and Analysis

- The framework API involved in project X's error message is defined as the framework API called by the client code closest to the Error message.
- Traceback shows the traceback message of `cnntext-classification-tf`. We can see that the TensorFlow API `concat()` is called in the client code, which is the closest to the `TypeError` message. Thus, the framework API that caused the project to crash is `concat()`.

```

Traceback (most recent call last):
  File "train.py", line 197, in <module>
    tf.app.run()
  File "/home/.../site-packages/tensorflow/python/platform/app.py"...
    sys.exit(main(sys.argv[1:] + flags_passthrough))
  File "train.py", line 194, in main
    train(x_train, y_train, vocab_processor, x_dev, y_dev)
  File "train.py", line 92, in train
    l2_reg_lambda=FLAGS.l2_reg_lambda)
  File "/home/.../cnn-text-classification-tf/text_cnn.py"...
    self.h_pool = tf.concat(pooled_outputs, 3)
  File "/home/.../site-packages/tensorflow/python/ops/array_ops.py"...
    dtype=dtypes.int32).get_shape()
  File "/home/.../site-packages/tensorflow/python/framework/ops.py"...
    ret = conversion_func(value, dtype=dtype, name=name, as_ref=as_ref)
  File "/home/.../site-packages/tensorflow/python/framework/constant_op.py"...
    return constant(v, dtype=dtype, name=name)
  File "/home/.../site-packages/tensorflow/python/framework/constant_op.py"...
    tensor_util.make_tensor_proto(value, dtype=dtype, shape=shape, verify_shape=verify_shape))
  File "/home/.../site-packages/tensorflow/python/framework/tensor_util.py"...
    _AssertCompatible(values, dtype)
  File "/home/.../site-packages/tensorflow/python/framework/tensor_util.py"...
    (dtype.name, repr(mismatch), type(mismatch).__name__))
TypeError: Expected int32, got list containing Tensors of type '_Message' instead.

```

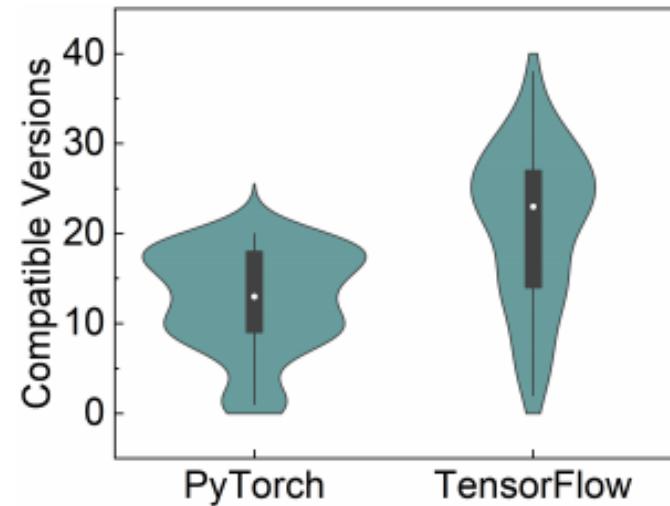
Client Framework Client Framework Error

API

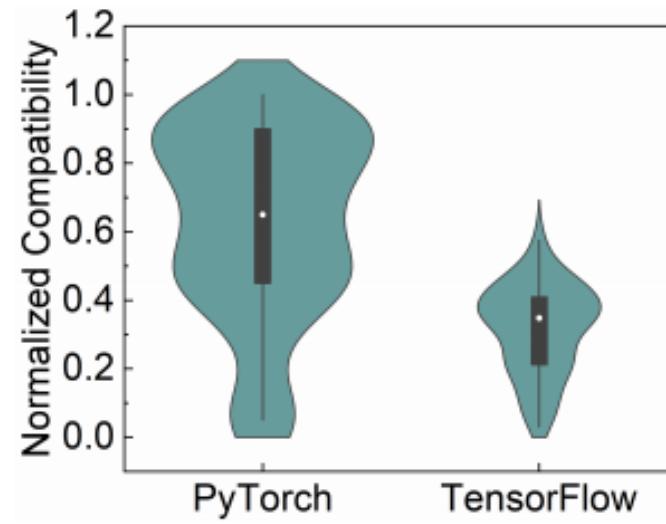
A red arrow points from the word "API" to the `tf.concat()` call in the client code. A red circle highlights the `tf.concat()` call in the client code.



RQ1: *Prevalence of Difference in Framework Version Compatibility Among DL Projects*



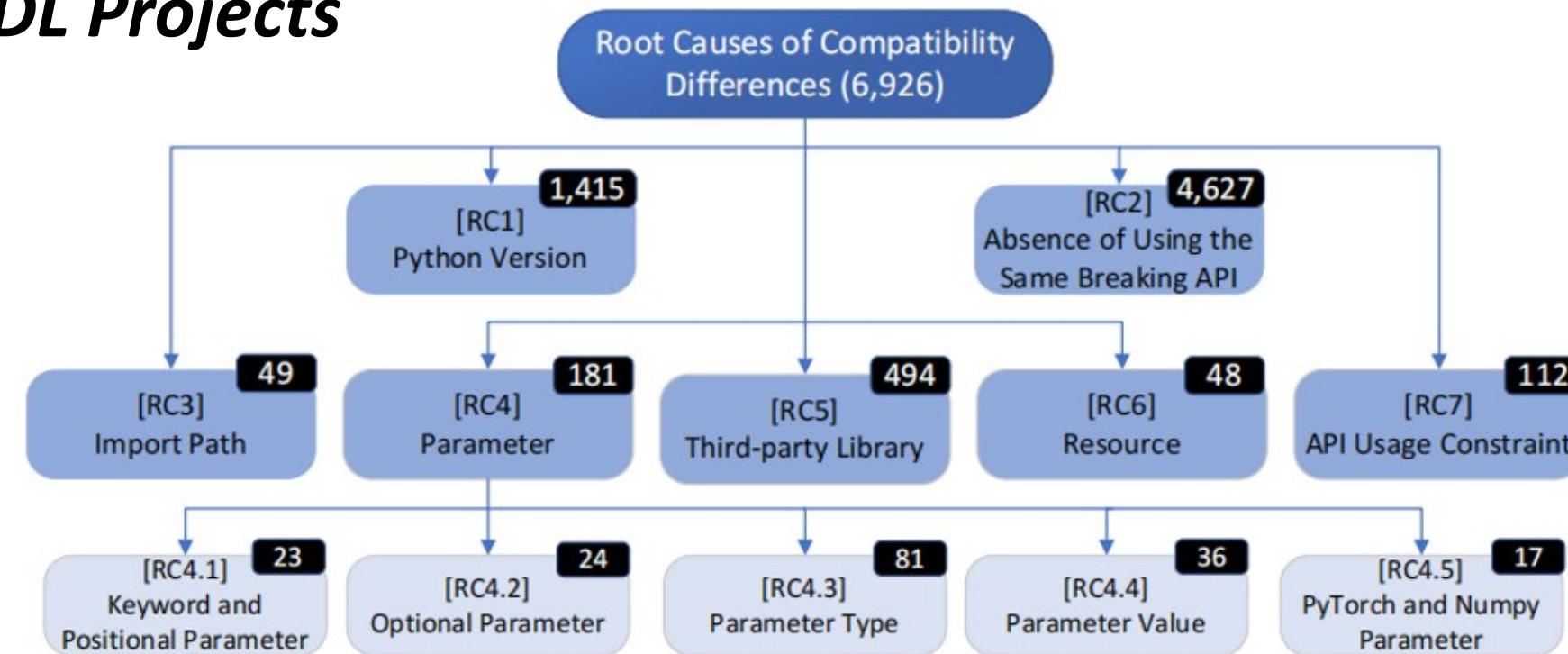
(a) Within-Framework



(b) Cross-Frameworks

Answer to RQ1: The difference in framework version compatibility is prevalent among DL projects. The number of compatible framework versions for the tested PyTorch projects ranges from 1 to 20, while the number for the tested TensorFlow projects ranges from 2 to 38. The framework version compatibility of PyTorch projects is better than that of TensorFlow projects.

RQ2: Root Causes of Difference in Framework Version Compatibility Among DL Projects



Answer to RQ2: The root causes of the difference in framework version compatibility among DL projects include **Python version, absence of using the same breaking API, import path, parameter, third-party library, resource, and API usage constraint.**



RC1: Python Version

Definition: DFVC is related to the Python version.

- DL framework: PyTorch
- Version: 1.6.0
- Python version: 3.5 and 3.7

```
Collecting torch==1.6.0
  Could not find a version that satisfies the requirement torch==1.6.0 (from versions: 1.0.0, 1.0.1, 1.0.1.post2, 1.1.0, 1.2.0, 1.3.0, 1.3.1, 1.4.0, 1.5.0, 1.5.1)
No matching distribution found for torch==1.6.0
```

	loading pretrained model from ./data/crnn.pth a----v--a-i-l-a-bb-l-e--- => available	Python 3.7	Normal
--	---	------------	--------

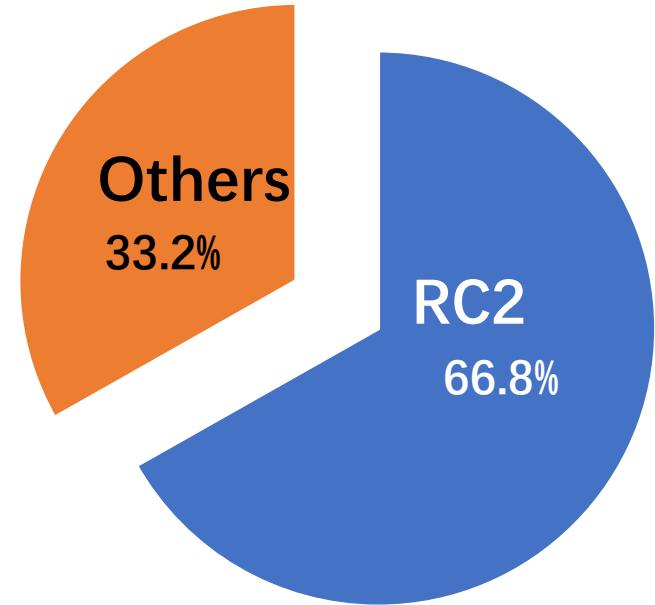
(a) AttentionWalk

(b) crnn.pytorch



RC2: Absence of Using the Same Breaking API

Definition: DFVC is related to the absence of using the same breaking API.





RC3: Import Path

Definition: DFVC is related to the API import path.

- DL framework: PyTorch
- Version: 1.9.0
- Import path: torch.utils.tensorboard and tensorboardX

```
myungsub/CAIN: main.py
...
11   from torch.utils.tensorboard import SummaryWriter
...
Traceback (most recent call last):
File "main.py", line 11, in <module>
  from torch.utils.tensorboard import SummaryWriter
File "/home/.../torch/utils/tensorboard/__init__.py", line 4, in <module>
  LooseVersion = distutils.version.LooseVersion
AttributeError: module 'distutils' has no attribute 'version'
```

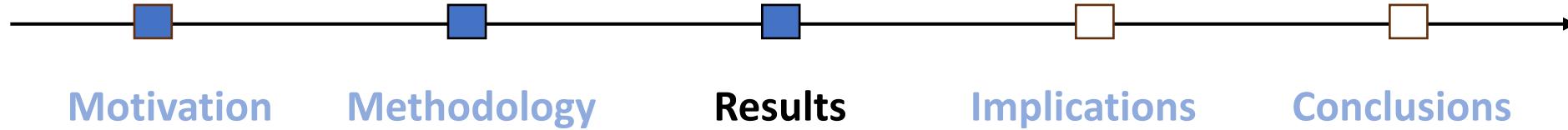
Crash

```
lufficc/SSD: trainer.py
...
64  try:
65    from torch.utils.tensorboard import SummaryWriter
66  except ImportError:
67    from tensorboardX import SummaryWriter
...
...
```

Normal

(a) CAIN

(b) SSD



RC4.1: Keyword and Positional Parameter

Definition: DFVC is related to the use of keyword and positional parameters.

- DL framework: **TensorFlow**
- Version: **1.13.2**
- Related API: **tf.nn.conv2d()**

```
githubharald/SimpleHTR: model.py
...
6   import tensorflow as tf
...
72  conv = tf.nn.conv2d(input=pool, filters=kernel, padding='SAME', strides=(1, 1, 1, 1))
...
Traceback (most recent call last):
File "main.py", line 202, in <module>
  main()
File "main.py", line 197, in main
  model = Model(char_list_from_file(), decoder_type, must_restore=True, dump=args.dump)
File "/home/.../SimpleHTR/src/model.py", line 43, in __init__
  self.setup_cnn()
File "/home/.../SimpleHTR/src/model.py", line 72, in setup_cnn
  conv = tf.nn.conv2d(input=pool, filters=kernel, padding='SAME', strides=(1, 1, 1, 1))
                                         ^~~~~~  
Crash
TypeError: conv2d() got an unexpected keyword argument 'filters'
```

(a) SimpleHTR

```
anishathalye/neural-style: vgg.py
...
5   import tensorflow.compat.v1 as tf
...
97 conv = tf.nn.conv2d(input, tf.constant(weights), strides=(1, 1, 1, 1), padding="SAME")
...
...
Iteration 1 / 1
content loss: 2.0134e+06
style loss: 6.78778e+07
tv loss: 23257.1
total loss: 6.99145e+07  
Normal
```

(b) neural-style



RC4.2: Optional Parameter

Definition: DFVC is related to whether the optional parameter is used.

- DL framework: **TensorFlow**
- Version: **1.4.1**
- Related API: **tf.reduce_max()**

```
weichen582/RetinexNet: model.py
...
8   import tensorflow as tf
...
17  input_max = tf.reduce_max(input_im, axis=3, keepdims=True)
...
...
Traceback (most recent call last):
File "/home/.../RetinexNet/model.py", line 17, in DecomNet
    input_max = tf.reduce_max(input_im, axis=3, keepdims=True)
TypeError: reduce_max() got an unexpected keyword argument 'keepdims'
```

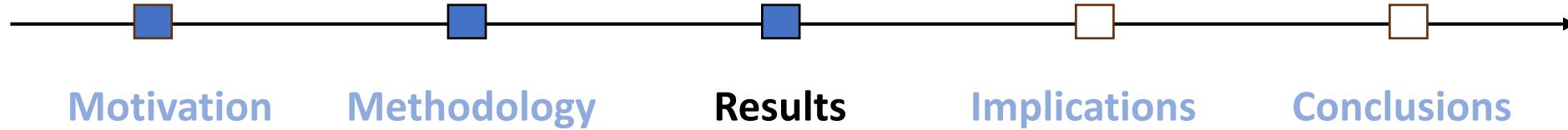
Crash

(a) RetinexNet

```
wizyoung/YOLOv3_TensorFlow: model.py
...
7   import tensorflow as tf
...
230 best_iou = tf.reduce_max(iou, axis=-1)
...
...
Init with stored values from ../model/snapshot-1
Recognized: "word"
Probability: 0.9937934279441833
```

Normal

(b) YOLOv3_TensorFlow



RC4.3: Parameter Type

Definition: DFVC is related to parameter types.

- DL framework: **TensorFlow**
- Version: **1.14.0**
- Related API: **tf.random_truncated_normal()**

```
lengstrom/fast-style-transfer: transform.py
1 import tensorflow as tf, pdb
...
66 weights_init = tf.Variable(tf.random.truncated_normal(weights_shape,
stddev=WEIGHTS_INIT_STDEV, seed=1), dtype=tf.float32)
...
...
Traceback (most recent call last):
...
File "src/transform.py", line 66, in _conv_init_vars
weights_init = tf.Variable(tf.random.truncated_normal(weights_shape,
stddev=WEIGHTS_INIT_STDEV, seed=1), dtype=tf.float32)
File "/home/l.../site-packages/tensorflow/python/ops/random_ops.py", line 173, in
truncated_normal
...
TypeError: Failed to convert object of type <class 'list'> to Tensor. Contents: [9, 9,
Dimension(3), 32]. Consider casting elements to a supported type.
```

(a) fast-style-transfer

```
githubbarald/SimpleHTR: model.py
...
6 import tensorflow as tf
...
69 kernel = tf.Variable(
70 tf.random.truncated_normal([kernel_vals[i], kernel_vals[i], feature_vals[i], feature_vals[i + 1]],
71 stddev=0.1))
...
...
Recognized: "word"
Probability: 0.9936366677284241
```

Normal

(b) SimpleHTR



RC4.4: Parameter Value

Definition: DFVC is related to parameter values.

- DL framework: **TensorFlow**
- Version: **1.0.1**
- Related API: **torch.nn.functional.interpolate()**

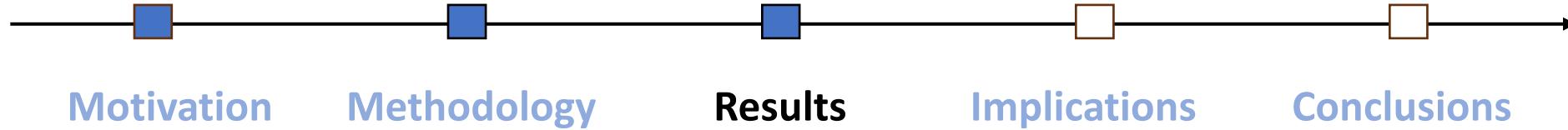
```
tohinz/ConSinGAN: models.py
...
2   import torch
...
29  x_up = torch.nn.functional.interpolate(x, size=size, mode='bicubic', align_corners=True)
...
Traceback (most recent call last):
...
File "/home/.../ConSinGAN/ConSinGAN/models.py", line 29, in upsample
    x_up = torch.nn.functional.interpolate(x, size=size, mode='bicubic', align_corners=True)
File "/home/.../site-packages/torch/nn/functional.py", line 2459, in interpolate
    " (got {})".format(input.dim(), mode))
NotImplementedError: Input Error: Only 3D, 4D and 5D input Tensors supported (got 4D) for
the modes: nearest | linear | bilinear | trilinear (got bicubic)
```

(a) ConSinGAN

```
jeya-maria-jose/KiU-Net-pytorch: ae.py
...
11  import torch.nn.functional as F
...
41  out = F.relu(F.interpolate(self.decoder3(out), scale_factor=(2,2), mode='bilinear'))
...
...
Total_params: 291234
epoch [0/1], loss:0.6605
```

Normal

(b) KiU-Net-pytorch



RC4.5: PyTorch and Numpy Parameter

Definition: DFVC is related to using the PyTorch and Numpy parameters.

- DL framework: **PyTorch**
- Version: **1.1.0**
- Related API: **torch.sum()**

```
polarisZhao/PFLD-pytorch: loss.py
1  import torch
...
13 weight_angle = torch.sum(1 - torch.cos(angle - euler_angle_gt), axis=1)
...
Traceback (most recent call last):
...
File "/home/.../torch/nn/modules/module.py", line 493, in __call__
    result = self.forward(*input, **kwargs)
File "/home/.../PFLD-pytorch/pfld/loss.py", line 13, in forward
    weight_angle = torch.sum(1 - torch.cos(angle - euler_angle_gt), axis=1)
TypeError: sum() received an invalid combination of arguments - got (Tensor, axis=int), but
...
```

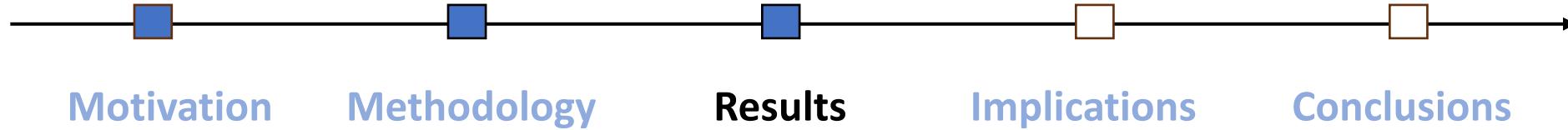
Crash

```
lukasruff/Deep-SAD-PyTorch: distributions.py
...
2  import torch
...
39 cross_entropy = torch.sum(p * torch.log(prior + eps), dim=1)
...
...
INFO:root:Finished training.
INFO:root:Starting testing...
INFO:root:Test Loss: 1.576255
INFO:root:Test AUC: 82.45%
INFO:root:Test Time: 0.678s
INFO:root:Finished testing.
```

Normal

(a) PFLD-pytorch

(b) Deep-SAD-PyTorch



RC5: Third-party Library

Definition: DFVC is related to third-party libraries and their versions.

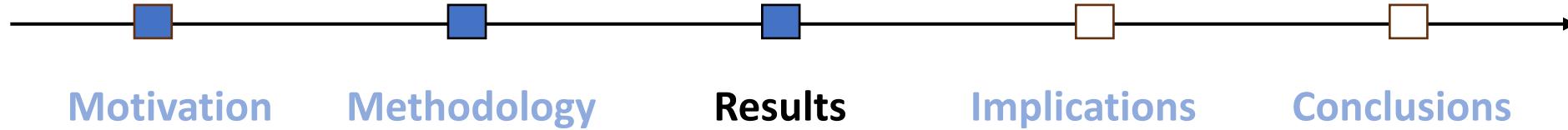
- DL framework: **TensorFlow**
- Version: **1.10.0**
- Related third-party library and its version: **matplotlib 3.3.4** and **2.1.0**

```
ywpkwon/siamese_tf_mnist: visualize.py
1   from tensorflow.examples.tutorials.mnist import input_data
...
4   import matplotlib.pyplot as plt 1.10.0 3.3.4
...
Traceback (most recent call last):
File "run.py", line 22, in <module>
    import visualize
File "/home/.../tensorflow/1.5/siamese_tf_mnist/visualize.py", line 4, in <module>
    import matplotlib.pyplot as plt
File "/home/.../site-packages/matplotlib/__init__.py", line 174, in <module> Crash
    _check_versions()
File "/home/.../site-packages/matplotlib/__init__.py", line 171, in _check_versions
    .format(modname, minver, module.__version__))
ImportError: Matplotlib requires numpy>=1.15; you have 1.14.5
```

(a) siamese_tf_mnist

```
nickliqian/cnn_captcha: train_model.py
...
4   import tensorflow as tf 1.10.0
...
6   import matplotlib.pyplot as plt 2.1.0
...
...
...loss 0.1267512441
...loss 0.1267512441
...
Normal
```

(b) cnn_captcha



RC6: Resource

Definition: DFVC is related to computing resources.

- DL framework: PyTorch
- Version: 1.1.0
- Related API: `torch.matmul()`

```
AlexHex7/Non-local_pytorch: non_local_embedded_gaussian.py
1 import torch
...
85 f = torch.matmul(theta_x, phi_x)
...
Traceback (most recent call last):
File "demo_MNIST_train.py", line 40, in <module>
predict = net(img_batch)
...
File "/home/.../torch/nn/modules/module.py", line 493, in __call__
result = self.forward(*input, **kwargs)
File "/home/.../Non-local_pytorch/lib/non_local_embedded_gaussian.py", line 85, in forward
f = torch.matmul(theta_x, phi_x)
RuntimeError: cublas runtime error : the GPU program failed to execute at /pytorch/aten/src/
THC/THCBlas.cu:450
```

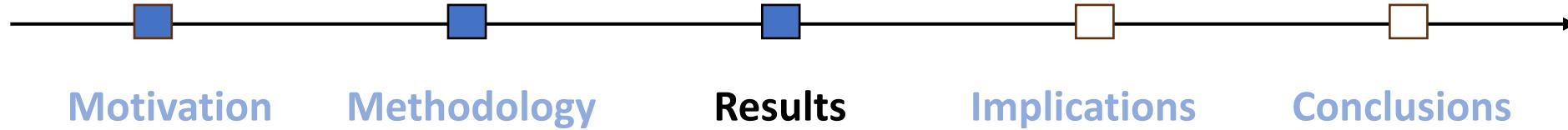
Crash

(a) Non-local_pytorch

```
benedekrozemberczki/SimGNN: demo.py
...
3 import torch
...
37 global_context = torch.mean(torch.matmul(embedding, self.weight_matrix), dim=0)
...
...
Baseline error: 0.41597. Model test error: 0.99222.
Model test error: 0.99222.
```

Normal

(b) SimGNN



RC7: Usage Constraint

Definition: DFVC is related to API usage constraints.

- DL framework: PyTorch
- Version: 1.7.0
- Related API: `view()`

```
YU1ut/MixMatch-pytorch: eval.py
1 import torch
...
16 correct_k = correct[:k].view(-1).float().sum(0)
...
Traceback (most recent call last):
File "train.py", line 417, in <module>
    main()
File "train.py", line 147, in main
    _train_acc = validate(labeled_trainloader, ema_model, criterion, epoch, use_cuda,
mode='Train Stats')
File "train.py", line 329, in validate
    prec1, prec5 = accuracy(outputs, targets, topk=(1, 5))
File "/home/.../MixMatch-pytorch/utils/eval.py", line 16, in accuracy
    correct_k = correct[:k].view(-1).float().sum(0)
RuntimeError: view size is not compatible with input tensor's size and stride (at least one dimension spans across two contiguous subspaces). Use .reshape(...) instead.
```

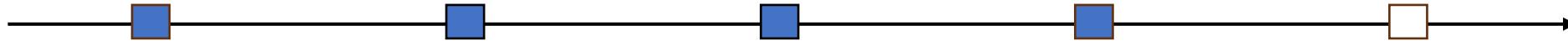
Crash

```
meijieru/crnn.pytorch: demo.py
1 import torch
...
34 preds = preds.transpose(1, 0).contiguous().view(-1)
...
loading pretrained model from ./data/crnn.pth
a----v--a-i-l-a-bb-l-e--- => available
```

Normal

(b) crnn.pytorch

(a) MixMatch-pytorch



Motivation

Methodology

Results

Implications

Conclusions



Implications for DL Practitioners

Implication #1: PyTorch exhibits better framework version compatibility compared to TensorFlow. Besides, it is recommended to choose a specific Python version, which is compatible with more framework versions



Python version	...	3.5	3.6	3.7	3.8	...
Compatible versions		41	60	43	25	
Compatible versions		9	19	20	14	

Implication #2: Developers should be mindful of potential changes in API import paths when switching between different framework versions.

Implication #3: When upgrading or downgrading DL framework versions, developers should pay attention to the changes in associated third-party libraries.



Motivation

Methodology

Results

Implications

Conclusions



Implications for DL Practitioners

Implication #4: Developers are suggested to utilize a try and except statement to catch exceptions when encountering compatibility issues after upgrading or downgrading the framework version.

```
try:  
    from torch.utils.tensorboard import SummaryWriter  
except ImportError:  
    from tensorboardX import SummaryWriter
```

Implication #5: When upgrading or downgrading a DL framework, it is crucial to carefully examine whether the APIs used comply with the requirements of the new framework version. In particular, developers should pay attention to the parameters.

Implication #6: To improve the framework version compatibility of DL projects, developers should be mindful of API usage constraints or use static valueflow analysis to reason about the constraints.



Motivation

Methodology

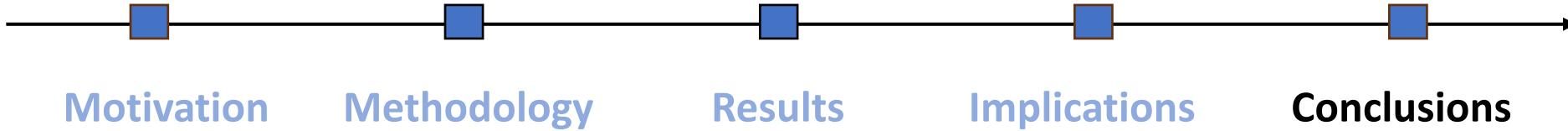
Results

Implications

Conclusions

Contributions

- We conduct the first empirical study on the DFVC among DL projects
- Seven root causes of the DFVC among DL projects
- Six implications to DL researchers and practitioners according to our findings
- Released the research data (<https://doi.org/10.5281/zenodo.8266949>)



Future Work

- Collect and test more DL projects and framework versions
- Develop tools to make DL projects compatible with more framework versions

Thanks!

Q&A

Guanping Xiao

Email: gpxiao@nuaa.edu.cn

Homepage: <https://guanpingxiao.github.io/>