Task-Driven Domain-Agnostic Learning with Information Bottleneck for Autonomous Steering

Yu Shen, Laura Zheng,
Tianyi Zhou, Ming C. Lin

Paper ID: 3254

Motivation

In autonomous driving, when we encounter

- Limited data in a target domain (e.g., a new environment), rich data in known domains
- Limited data in real-world scenarios (e.g., accident data), rich in simulator

How?

Insights

- What information from other domains is useful to target domain task?
- What information from target domain is not contained in the previous answer but is potentially useful?

Insights

- What information from other domains is useful to target domain task?
  - ***Domain-invariant Features***

- What information from target domain is not contained in the previous answer but is potentially useful?
  - ***Domain-specific Features***

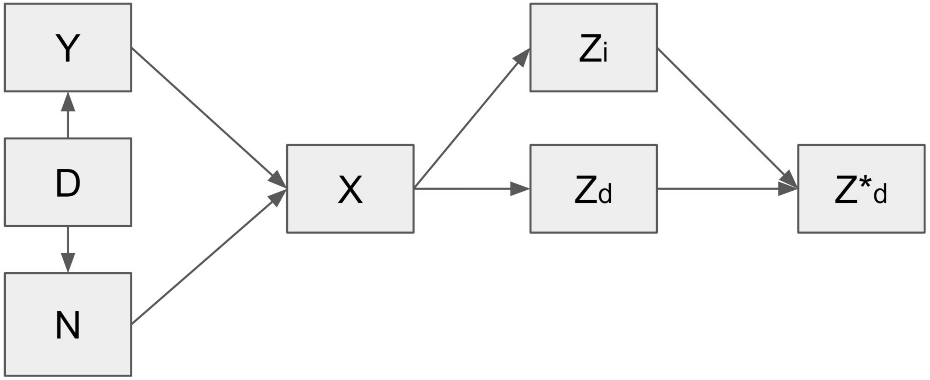
Insights

- What information from other domains is useful to target domain task?
    - ***Domain-invariant Features***
- What information from target domain is not contained in the previous answer but is potentially useful?
    - ***Domain-specific Features***

How to utilize both?

Information Theory: Causal Graph

X	Input data variable
Y	Task label variable
D	Domain variable
N	Nuisance variable
Z_i	Domain-invariant latent variable
Z_d	Domain-dependent latent variable
Z^*_d	Combined latent variable



Loss

Learning objective:

$$\max_{\theta_{g_i},\theta_{g_d},\theta_{f_{d^*}}} I(Z_d^*, Y) - \lambda I(Y, D | Z_d^*)$$

Reformed objective:

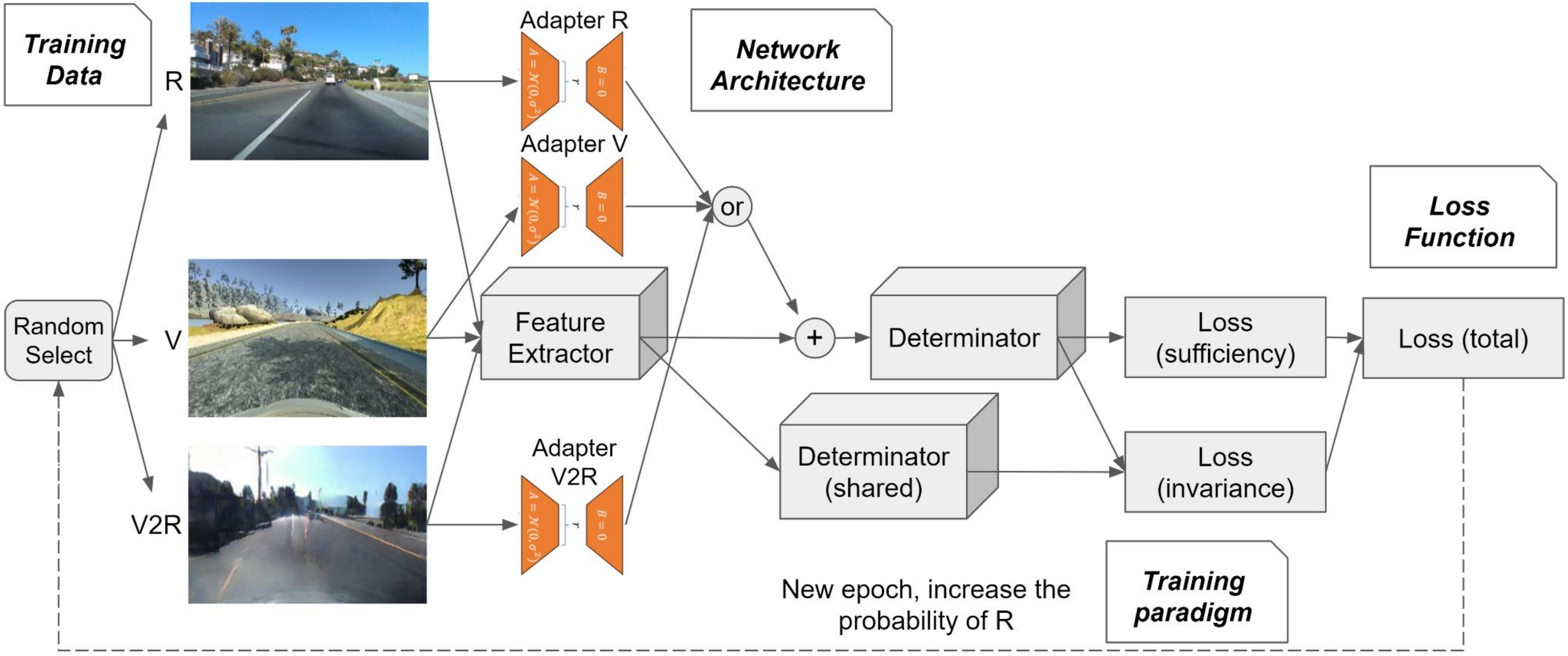
$$\min_{g_i, g_d, f_{d^*}} \max_{f_i} \underbrace{\mathcal{L}_{d^*}(g_i, g_d, f_{d^*})}_{\text{Sufficiency Loss}} + \lambda \underbrace{(\mathcal{L}_{d^*}(g_i, g_d, f_{d^*}) - \mathcal{L}_i(g_i, f_i))}_{\text{Invariance Loss}}$$

Where:

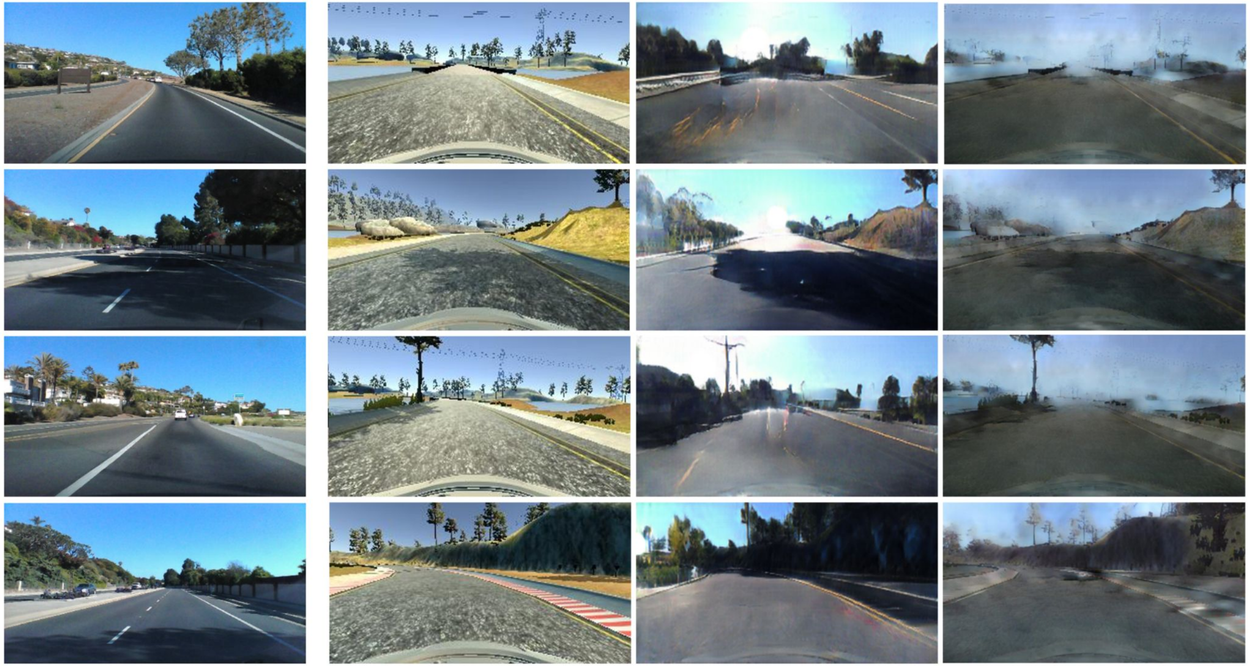
$$\mathcal{L}_{d^*} = \mathbf{E}_{x,y}[L(y, f_{d^*}(g_i(x), g_d(x)))]$$

$$\mathcal{L}_i = \mathbf{E}_{x,y}[L(y, f_i(g_i(x)))]$$

Our Method


Training Data
R
V
V2R
Random Select
Adapter R
Adapter V
Adapter V2R
Network Architecture
or
Feature Extractor
+
Determinator
Determinator (shared)
Loss (sufficiency)
Loss (invariance)
Loss (total)
Loss Function
New epoch, increase the probability of R
Training paradigm

Dataset Images



(a) Real (R) (b) Virtual (V) (c) CycleGAN (T_C) (d) MUNIT (T_M)

Fig. 3: Sample images of various datasets. (a) the SullyChen dataset [7] (real dataset, denoted by R). (b) the Udacity dataset [1] (virtual dataset, denoted by V). (c) style-transferred images from virtual to real using CycleGAN [47] (denoted by T_C). (d) style-transferred images from virtual to real using MUNIT [21] (denoted by T_M).

Experiments: Comparison with Other Methods

ICRA2024

TABLE IV: Accuracy comparison with domain-adaptation & task-adaptation methods. Ours outperforms others with highest accuracy (mAcc) & lowest mean square error (MSE).

		$\mathrm{MA}_R(M)$ (%) on different angle threshold τ (degree)						
	Method	$\tau = 1.5$	$\tau = 3.0$	$\tau = 7.5$	$\tau = 15$	$\tau = 30$	**mAcc**	**MSE**
	Baseline	59.5%	82.1%	93.9%	96.3%	98.6%	86.04%	0.96
(a) Domain Adaptation	DANN [13]	28.9%	52.5%	79.3%	92.2%	97.3%	70.04%	0.58
	ADDA [40]	33.6%	54.3%	84.4%	93.2%	97.5%	72.6%	0.43
	BSP [9]	38.9%	60.4%	87.5%	95.1%	98.4%	76.06%	0.32
(b) Task Adaptation	DELTA [30]	61.9%	80.9%	93.9%	97.7%	99.2%	86.72%	0.16
	BSS [8]	67.0%	83.4%	93.8%	97.5%	98.8%	88.1%	0.21
	StochNorm [26]	53.7%	78.5%	92.8%	97.3%	99.2%	84.3%	0.18
	Ours	**70.5%**	**84.3%**	**93.8%**	**97.9%**	**99.4%**	**89.2%**	**0.15**

Experiments: Comparison with Other Methods

TABLE V: Mean Accuracy comparison with domain adaptation and task adaptation methods on different metrics. Our method outperforms others under nearly all angle thresholds.

		$\mathrm{MA}_R(M)$ (%) on different angle threshold $\tau = 3(y_{gt}/30)^{1/\alpha}$ (degree)					
	Method	$\alpha = 1 (\tau = 0.1 y_{gt})$	$\alpha = 2$	$\alpha = 5$	$\alpha = 10$	$\alpha = \infty (\tau = 3)$	**mAcc**
	Baseline	31.4%	50.0%	69.7%	74.8%	79.7%	61.13%
(a) Domain Adaptation	DANN [13]	11.3%	22.9%	37.3%	44.9%	51.4%	33.55%
	ADDA [40]	11.9%	20.7%	38.5%	46.1%	52.3%	33.91%
	BSP [9]	17.0%	31.8%	44.9%	52.0%	58.6%	40.86%
(b) Task Adaptation	DELTA [30]	33.6%	56.2%	72.3%	76.6%	79.3%	63.59%
	BSS [8]	36.3%	**63.1%**	74.6%	78.3%	81.6%	66.80%
	StochNorm [26]	29.5%	43.8%	63.7%	71.7%	76.8%	57.07%
	Ours	**36.7%**	60.4%	**77.5%**	**80.5%**	**82.6%**	**67.54%**

Experiments: Comparison with Other Methods

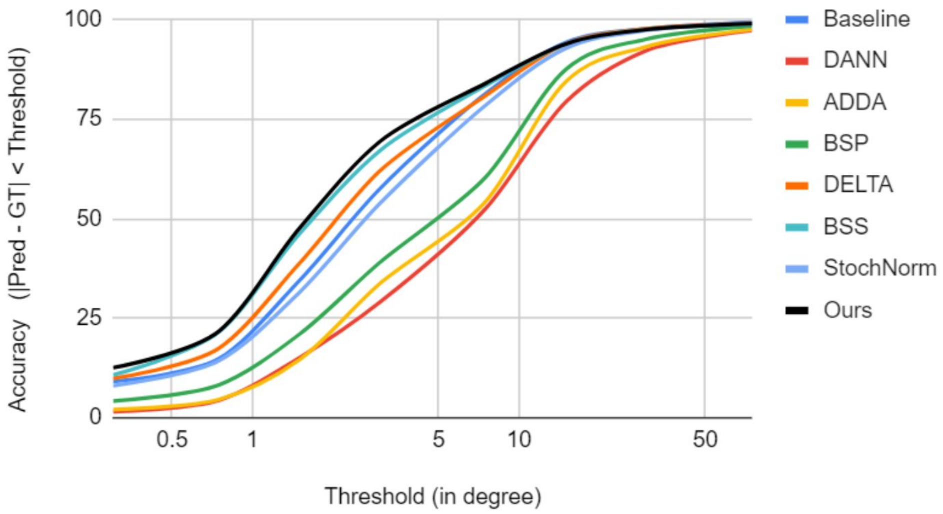


Fig. 4: **Threshold-Accuracy Curve**. Our method (in black) achieves the best (highest) performance – above all other methods.

Experiments: Domain

ICRA2024

Experiments show the existence of domain gap and domain-invariant information.

TABLE I: Mean Accuracy cross comparison. RV stands for transferring real dataset to virtual style, VR for transferring virtual dataset to real style. $CGAN$ for the Cycle-GAN method, and CR for the color remapping method.

Test	Train: R	V	RV_{CGAN}	VR_{CGAN}	RV_{CR}	VR_{CR}
R	**88.36%**	*31.16%*	**48.83%**	26.87%	**70.17%**	30.08%
RV_{CGAN}	**51.42%**	34.22%	**80.08%**	29.34%	**53.18%**	*38.86%*
RV_{CR}	**60.89%**	35.86%	**48.18%**	27.79%	**85.50%**	*37.41%*

Experiments: Training Paradigm

Experiments to help choose the best training paradigm.

TABLE II: Mean Accuracy comparison with different training paradigms. From (a) we can verify the existence of a domain gap between the virtual, style-transferred and real datasets. From (b,c,d,e,f), we find that (e) **"finetuning with reinitialization" outperforms other training paradigms**.

	Model (M)	$\text{MA}_R(M)$
(a) Single dataset	train(R)	88.36%
	train($R1$)	32.02%
	train(V)	31.16%
	train(T_C)	26.87%
	train(T_M)	25.56%
(b) Simply combine	train($R1$ + R)	82.32%
	train(V + R)	75.74%
	train(T_C + R)	75.44%
	train(T_M + R)	76.85%
(c) Finetuning	train($R1$) $\rightarrow$ train(R)	81.93%
	train(V) $\rightarrow$ train(R)	83.54%
	train(T_C) $\rightarrow$ train(R)	82.70%
	train(T_M) $\rightarrow$ train(R)	79.04%
(d) Partially finetuning	train($R1$) $\rightarrow$ ptrain(R)	70.86%
	train(V) $\rightarrow$ ptrain(R)	73.66%
	train(T_C) $\rightarrow$ ptrain(R)	77.17%
	train(T_M) $\rightarrow$ ptrain(R)	72.97%
(e) Finetuning with reinitialization	train($R1$) $\rightarrow$ train(R)	**88.71%**
	train(V) $\rightarrow$ train(R)	**87.50%**
	train(T_C) $\rightarrow$ train(R)	**83.12%**
	train(T_M) $\rightarrow$ train(R)	**80.26%**
(f) Partially finetuning with reinitialization	train($R1$) $\rightarrow$ ptrain(R)	76.94%
	train(V) $\rightarrow$ ptrain(R)	75.08%
	train(T_C) $\rightarrow$ ptrain(R)	77.78%
	train(T_M) $\rightarrow$ ptrain(R)	74.28%

Experiments: Architecture Component

ICRA2024

Experiments to help choose the best architecture component to decouple domain-invariant and domain-specific features.

TABLE III: Mean Accuracy (MA) comparison with different network architectures. **Adapter achieves best MA.**

	M	$\mathrm{MA}_R(M)$
(a) Finetuning + reinit header + BN	$\mathtt{train}(V) \rightarrow \mathtt{train}(R)$ $\mathtt{train}(R1) \rightarrow \mathtt{train}(R)$	80.53% 80.77%
(b) AdvProp BN	$\mathtt{train}(R, V)$ $\mathtt{train}(R, R1)$	71.22% 75.83%
(c) Finetuning + reinit header + adapter	$\mathtt{train}(V) \rightarrow \mathtt{train}(R)$ $\mathtt{train}(R1) \rightarrow \mathtt{train}(R)$	**81.32%** **82.71%**

Experiments: Ablation

TABLE VIII: Ablation study. ADP for adapter, STB for style transferred branch, DP for dynamic probability in each domain, and IBL for information bottleneck loss.

	$\mathrm{MA}_R(M)$ (%) on different angle threshold τ (degree)						
Method	$\tau = 1.5$	$\tau = 3.0$	$\tau = 7.5$	$\tau = 15$	$\tau = 30$	**mAcc**	**MSE**
Baseline	59.5%	82.1%	93.9%	96.3%	98.6%	86.04%	1.96
Ours w/o ADP	58.4%	80.3%	93.4%	97.7%	98.6%	85.68%	0.19
Ours w/o STB	68.0%	81.6%	94.1%	97.7%	99.0%	88.08%	0.16
Ours w/o DP	65.6%	82.2%	93.4%	97.3%	98.8%	87.46%	0.18
Ours w/o IBL	69.3%	84.0%	**93.9%**	97.5%	99.0%	88.74%	0.18
Ours	**70.5%**	**84.3%**	93.8%	**97.9%**	**99.4%**	**89.2%**	**0.15**

Experiments: Datasets and Backbones

TABLE VI: Comparison on different datasets.

		$\text{MA}_R(M)$ (%) on different angle threshold τ (degree)						
	Method	$\tau = 1.5$	$\tau = 3.0$	$\tau = 7.5$	$\tau = 15$	$\tau = 30$	**mAcc**	**MSE**
(a) SullyChen	BSS [8]	67.0%	83.4%	93.8%	97.5%	98.8%	88.1%	0.21
	Ours	70.5%	84.3%	93.8%	97.9%	99.4%	**89.2%**	**0.15**
(b) Audi	BSS [8]	59.5%	72.3%	81.6%	86.9%	89.7%	78%	0.88
	ours	62.5%	75.3%	84.8%	89.1%	92.4%	**80.8%**	**0.65**
(c) Honda	BSS [8]	55.4%	70.9%	77.8%	83.7%	86.5%	74.86%	1.16
	ours	57.6%	73.9%	80.2%	85.7%	89.1%	**77.3%**	**0.91**

TABLE VII: Comparison on different backbones.

		$\text{MA}_R(M)$ (%) on different angle threshold τ (degree)						
	Method	$\tau = 1.5$	$\tau = 3.0$	$\tau = 7.5$	$\tau = 15$	$\tau = 30$	**mAcc**	**MSE**
(a) PilotNet	BSS [8]	67.0%	83.4%	93.8%	97.5%	98.8%	88.1%	0.21
	Ours	70.5%	84.3%	93.8%	97.9%	99.4%	**89.2%**	**0.15**
(b) ResNet	BSS [8]	71.8%	84.9%	93.8%	97.4%	98.3%	89.24%	0.15
	ours	72.3%	85.6%	94.5%	98.2%	99.5%	**90.02%**	**0.13**
(c) LSTM	BSS [8]	73.1%	85.4%	94%	97.5%	98.9%	89.78%	0.14
	ours	74.5%	86.9%	95.1%	98.6%	99.7%	**90.96%**	**0.12**

Conclusion

- A novel framework for *domain-agnostic learning* in the end-to-end autonomous steering task, with
  - Loss: Information bottleneck loss
  - Architecture: Adapter for domain-specific feature extraction
  - Training paradigm: Dynamic probability for domain data selection
  - Training data: Style transferred branch for domain-agnostic feature decoupling

- Performance improvement
  - Up to **19.16%** compared to other domain adaptation methods
  - Up to **4.9%** compared to other task adaptation methods

Thank you!