

Incremental Adversarial Domain Adaptation for Continually Changing Environments

Markus Wulfmeier¹, Alex Bewley¹ and Ingmar Posner¹

Abstract— Continuous appearance shifts such as changes in weather and lighting conditions can impact the performance of deployed machine learning models. While unsupervised domain adaptation aims to address this challenge, current approaches do not utilise the continuity of the occurring shifts. In particular, many robotics applications exhibit these conditions and thus facilitate the potential to incrementally adapt a learnt model over minor shifts which integrate to massive differences over time. Our work presents an adversarial approach for lifelong, incremental domain adaptation which benefits from unsupervised alignment to a series of intermediate domains which successively diverge from the labelled source domain. We empirically demonstrate that our incremental approach improves handling of large appearance changes, *e.g.* day to night, on a traversable-path segmentation task compared with a direct, single alignment step approach. Furthermore, by approximating the feature distribution for the source domain with a generative adversarial network, the deployment module can be rendered fully independent of retaining potentially large amounts of the related source training data for only a minor reduction in performance.

I. INTRODUCTION

Appearance changes based on lighting, seasonal, and weather conditions provide a significant challenge for outdoor robots relying on machine learning models for perception. While providing high performance in their training domain, visual shifts occurring in the environment can result in significant deviations from the training distribution, severely reducing accuracy during deployment. Commonly, this challenge is partially counteracted by employing additional training methods to render these models invariant to their application domain [1].

For scenarios where labelled data is unavailable in the target domain, the problem can be addressed in the context of unsupervised domain adaptation [2], [3]. Recent state-of-the-art approaches which address this challenge operate by training deep neural networks within an adversarial domain adaptation (ADA) framework. These approaches are characterised by the optimisation of potentially multiple encoders with the objective to confuse a domain discriminator operating on their output [3], [4], [5] in addition to their main objective. The main intuition behind this framework is that by training the encoder to obtain a domain invariant embedding, we allow the main supervised task to be robust to changes in the application domain.

Recent successes based on adversarial domain adaptation have achieved state-of-the-art performance on toy datasets [3],

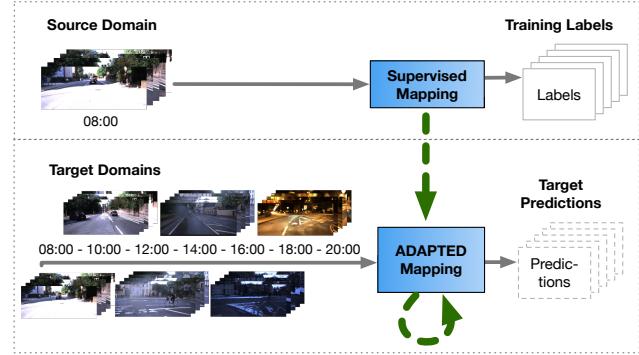


Fig. 1: Incremental Adversarial Domain Adaptation. Instead of performing domain adaptation over large shifts at once, IADA splits domain alignment into simpler subtasks. After adapting the feature embedding of the initial target domain, the approach incrementally refines all modules to the currently perceived target domains.

[6], [7], [8] as well as real-world applications for autonomous driving within changing environmental conditions [9], [5]. However, domains with a significant difference in appearance – such as day and night – continue to present a substantial challenge [5]. We conjecture that the observed change in environmental conditions in many domains of continuous deployment (*e.g.* in autonomous driving) is up to some extend composed via a gradual process which accumulates to produce massive differences over extended periods. This work exploits the incremental changes observed throughout deployment to continuously counteract the domain shift by updating discriminator and encoder incrementally while observing the visual shift (as illustrated in Figure 1).

During domain adaptation training, existing methods rely on data from the source domain to ensure training the discriminator based on a balanced distribution of source and target data to prevent overfitting to the target domain. These approaches therefore require the storage of potentially massive datasets, which can provide a challenge in particular for mobile applications with limited available memory.

Similar to work on synthetic dataset extension [10], we remove this requirement by training a generative adversarial network (GAN) [11] to imitate the marginal encoder feature distribution in the source domain. We empirically demonstrate that domain adaptation via aligning target encoder features with GAN generated samples instead of source domain feature embeddings only results in minor performance reduction. Crucially, this means that the deployment module is fully independent of the size of source domain dataset, enabling ap-

¹The authors are with the Applied Artificial Intelligence Lab, Oxford Robotics Institute, University of Oxford, United Kingdom; markus, bewley, ingmar@robots.ox.ac.uk

plication on mobile platforms with limited memory resources.

The approach is evaluated both on synthetic and real-world data. An artificial dataset is created with direct control over the number of intermediate domains and the strength of the incremental shifts for illustration and demonstration purposes. The following real-world evaluation focuses on a drivable-path segmentation task for autonomous driving based on segments from the Oxford RobotCar Dataset [12] with different illumination conditions from different times of day.

The contributions of our work are as follows:

- Introduction of a method for *incremental* unsupervised domain adaptation for platforms deployed in continuously changing environments.
- Presentation of an additional method to remove the requirement of retaining extensive amounts of source data by modelling the feature representation of source domain data with a generative model.
- Quantitative investigation of the influence of dividing the adaptation task into incremental alignment over smaller shifts based on a synthetic toy example.
- Application of the proposed method to the real-world task of drivable-terrain segmentation and proof of feasibility for online application in the context of runtime evaluation on an NVIDIA GPU.

II. RELATED WORK

Continuously changing environment appearances have been a long-standing challenge for robot deployment as shifts between training and deployment data can seriously degrade model performance. Considerable efforts have been focused on designing and comparing various feature transformations with the goal of creating representations invariant to environmental change [13]. Other approaches address the problem through retaining multiple experiences [14] or synthesising images between discrete domains [15]. However, it is unclear how these systems can efficiently scale to a continuous shift in the domain distribution.

In recent years, there has been a steady trend towards applying deep networks for various robotics tasks, where early layers act as a feature encoder with a supervised loss for the desired task on the output of the network. Unfortunately, even such powerful models still suffer from the problem of shifts in domain appearance. This has prompted a number of works which try to address this issue [3], [9], [16], [5].

The possibility to directly optimise complete feature representations via backpropagation for domain invariance [3] or target-source mappings [2] has lead to significant success of deep architectures in this field. Long *et al.* [2] focus on minimising the Maximum Mean Discrepancy for the feature distributions of multiple layers of the network architecture. Rozantsev *et al.* [17] extend in a similar direction and impose a penalty for deviations in the network parameters across domains. Sun *et al.* [18] align second order statistics of layer activations for source and target domains. Hoffman *et al.* [9] match the label statistics between the true source and predicted target labels for semantic classification.

Furthermore, adversarial approaches to domain adaptation have been introduced [3], [4], [6], which rely on training a domain discriminator to classify the domains underlying an encoder's feature distribution. While adversarial training techniques have been shown to be notoriously unstable and difficult to optimise, there has been a pronounced body of work towards improving their stability, including more dominant use of the confusion loss [11] and more recently the Wasserstein GAN framework [19].

All the above mentioned works treat the unsupervised domain adaptation problem as a batch transition without exploiting temporal coherence commonly available to robots in continuous deployment. Continuous refinement has however been actively researched in supervised learning for many years (*e.g.* [20], [21], [22]), yet there has been little work on methods for unsupervised domain adaptation. One notable exception is the work by Hoffman *et al.* [23], which addresses the problem with predefined features and focuses on the challenges of aligning to a continuously reshaping target domain. This work seeks to extend the recently developed approach of adversarial domain adaption to a continuously evolving target domain by capitalising on the perpetual observations made by a robot.

III. METHOD

Incremental Adversarial Domain Adaptation addresses the problem of degraded model performance due to continuously shifting environmental conditions. This includes changes caused by weather and lighting occurring in outdoor scenarios. Compared to the regular single-step domain adaptation paradigm, we benefit in applications building on continual deployment through exploitation of the incremental changes that integrate to large domain shifts. Continuously observable lighting or seasonal shifts in outdoor robotics and other applications constitute a prime example for this paradigm.

The approach extends adversarial domain adaptation approaches [3] aiming to facilitate learning a feature encoding f which is invariant with respect to the origin domain of its input data. In this way, the method enables the application of a supervised module trained only on source domain data to incoming unsupervised data from the application domain as depicted in Figure 2.

In comparison to existing methods [3], [7], [5] which frame the task of unsupervised domain adaptation as a one-step approach between distinct source and target domains, IADA treats the incoming data as a stream of incrementally changing domains. Exploiting access to data from these incremental changes facilitates alignment over greater overall shifts between the target and source domains. The encoder and discriminator models are updated gradually to enable alignment for each incrementally shifted domain.

Adversarial domain adaptation [3] generally tends to be hyperparameter search intensive [5] as it - in addition to the adversarial min-max problem - is affected by the potential conflict of the domain invariance objective and the supervised objective. Intuitively, by dividing the domain alignment procedure into smaller incremental shifts, we

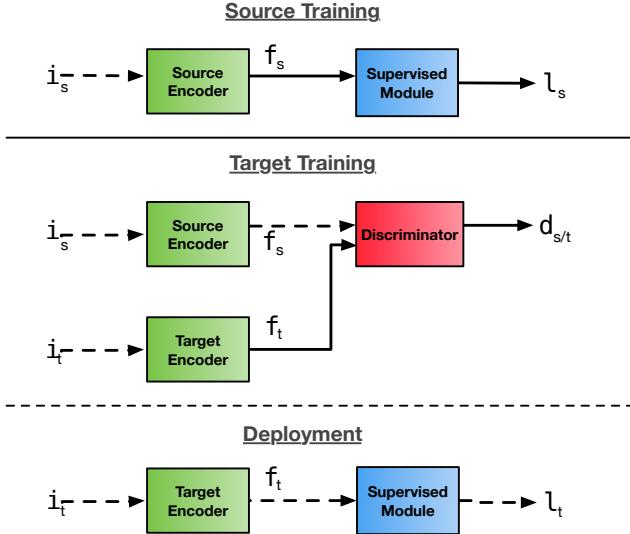


Fig. 2: Network architecture and information flow for IADA. After the optimisation of source encoder and supervised model, the target encoder is trained to confuse the domain discriminator, leading to domain invariant feature representations. During deployment, the target encoder is connected to the supervised module. Dotted arrows represent only forward passes while solid lines display forward and gradient backward pass.

simplify the overall task which can minimise the loss of relevant information.

The training procedure is split into two principal segments: *offline* supervised optimisation on source domain data and the unsupervised domain adaptation procedure, which potentially can be run *online* during platform deployment as displayed in Figure 2.

Hereinafter, let θ_X be the parametrisation of module X and i the incoming images. Source and target domains are represented with subscripts s and t respectively. The supervised training procedure optimises the supervised module S with the predicted label $l = S(f, \theta_S)$ as well as the source domain encoder E_s based on a supervised task (e.g. classification or segmentation as in Section IV).

The parameters of both of these modules remain unchanged during training for domain adaptation, which enables us to keep source performance unaffected (an approach suggested for regular ADA in [8]). Only the target encoder E_t and discriminator D are trained via their respective objectives \mathcal{L}_{E_t} and \mathcal{L}_D in Equations 1 and 2 to align the target and source encoder feature spaces. Let $f_s = E_s(i_s, \theta_{E_s})$ and $f_t = E_t(i_t, \theta_{E_t})$ respectively denote the feature encoding of source and target images i_s and i_t .

$$\mathcal{L}_{E_t}(\theta_{E_t}, \theta_D) = -\mathbb{E}_{i_t \sim T}[\log(D(f_t, \theta_D))] \quad (1)$$

$$\begin{aligned} \mathcal{L}_D(\theta_{E_s}, \theta_{E_t}, \theta_D) &= -\mathbb{E}_{i_s \sim S}[\log(D(f_s, \theta_D))] \\ &\quad -\mathbb{E}_{i_t \sim T}[\log(1 - D(f_t, \theta_D))] \end{aligned} \quad (2)$$

The target encoder weights are initialised with parameters from the source encoder trained on the supervised task. These

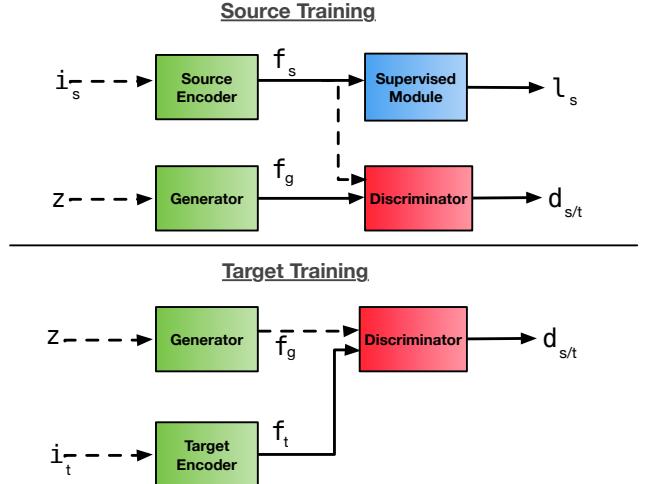


Fig. 3: Network architecture and information flow for training with a generative model approximating the marginal source feature distribution. The approach additionally trains a GAN during the source training procedure but does not propagate gradients for the adversarial loss to the source encoder to ensure unmodified source domain performance. Subsequently the target encoder is trained to mimic the feature distribution of the - now fixed - GAN. Dotted arrows represent only forward passes while solid lines display forward and gradient backward pass.

inchoate parameters are then subsequently adapted to align to the currently encountered target data by optimising both the target encoder and discriminator using the objectives in Equations 1 and 2 respectively. Intuitively, this procedure entails using the optimised parameters from the previously encountered target domain as initialisation for adapting to the current domain. The currently encountered unsupervised data is hereby utilised to fill a buffer from which is continuously sampled for the domain adaption training procedure.

A. Source Distribution Modelling

As IADA's benefits apply in the context of continuously deployed platforms, the inherent requirement of ADA-based methods to retain potentially large amounts of source training data can constrain its application. Limitations on the weight and size of platforms often leads to restrictions on computational resources including data storage. To counteract this requirement, we additionally extend our method with a GAN-based approach to mimic the source domain's feature distribution, thus rendering the approach independent of the amount of source data during the domain adaptation task.

More concretely, we optimise a generator G which maps from n -dimensional, normally distributed noise $z \sim \mathcal{N}(\mu = 0, \sigma = 1)$ to approximate the feature distribution in our source domain during the offline training step. While the original GAN framework [11] aims at mimicking natural images, our approach simply aims to imitate the feature encoding of images (displayed in Figure 3). The resulting objectives for generator \mathcal{L}_G and discriminator \mathcal{L}_D are displayed in

Equations 3 and 4. Let f_g denote the generator features generated as $f_g = G(z, \theta_G)$.

$$\mathcal{L}_G(\theta_G, \theta_D) = -\mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma)}[\log(D(f_g, \theta_D))] \quad (3)$$

$$\begin{aligned} \mathcal{L}_D(\theta_G, \theta_{E_s}, \theta_D) &= -\mathbb{E}_{i \sim S}[\log(D(f_s, \theta_D))] \\ &\quad -\mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma)}[\log(1 - D(f_g, \theta_D))] \end{aligned} \quad (4)$$

Subsequently during the domain adaptation procedure, the target encoder is optimised to align to the feature distribution of the GAN, whose parameters remain static to model the source domain. Instead of optimising the discriminator to classify between source and target domain, in this scenario it learns to distinguish between synthetically generated source features and actual target features encoding target images. Target encoder and discriminator are optimised towards the objectives \mathcal{L}_{E_t} and \mathcal{L}_D in Equations 5 and 6 respectively.

$$\begin{aligned} \mathcal{L}_{E_t}(\theta_{E_t}, \theta_D) &= -\mathbb{E}_{i \sim T}[\log(D(f_t, \theta_D))] \quad (5) \\ \mathcal{L}_D(\theta_G, \theta_{E_t}, \theta_D) &= -\mathbb{E}_{z \sim \mathcal{N}(\mu, \sigma)}[\log(D(f_g, \theta_D))] \\ &\quad -\mathbb{E}_{i \sim T}[\log(1 - D(f_t, \theta_D))] \end{aligned} \quad (6)$$

Similar to IADA, we utilise all models, which are trained in the source domain, as initialisation for training in the target domain. For SDM, this procedure additionally includes the discriminator. Lastly, the deployment setup is equivalent to IADA.

IV. EXPERIMENTS

Our evaluation is split into two parts: we first investigate a toy scenario with artificially induced domain shift for the purpose of visualisation and clarification, then we demonstrate performance gains in a continuous deployment scenario for drivable-path segmentation for autonomous mobility.

The evaluation compares IADA against its one-step counterpart ADA, and furthermore investigates the influence of source domain modelling based on Section III-A. The evaluation metric depends on the supervised task in the respective target domains, classification accuracy for the toy example and mean average precision for the drivable-path segmentation task.

While ADA only utilises the final source domain, IADA has access to all incremental domains. To evaluate if the cause of IADA's advantages simply is the reliance on a larger dataset, we additionally introduce ADA Union. The method combines all target domains into a single dataset and performs regular ADA with respect to this union over all target domains.

A. Toy Example: Incrementally Transformed MNIST

To quantify the benefits of IADA in relation to the strength of domain shifts and the number of intermediate domains, we first evaluate the approach in a scenario based on increasingly, synthetically deformed versions of the popular MNIST dataset.

We create additional, transformed copies of the original dataset with height-rescaled digits of between 0.9 to 0.5

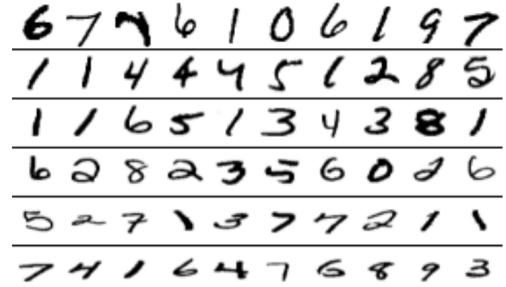


Fig. 4: Incremental deformation of MNIST digits from full to half height over 5 intermediate domains. Top row: original source data. Bottom row: maximally transformed target domain.

times the original height, which are visualised in Figure 4. These synthetically transformed domains enable us to create a scenario with full control over the underlying domain shift and ensure that the occurring changes can be observed and utilised for domain adaptation in arbitrary detail.

We employ a Network-in-Network like architecture [24] with exponential linear activation functions [25] splitting after the last hidden layer and applying a discriminator with 2 hidden layers and each 512 neurons. All parameters before the split are duplicated for source and target encoders while the supervised module consists of the last fully connected layer. The adversarial loss is weighted by a factor of 0.001 for domain adaptation as well as training the GAN as part of the source domain modelling step.

target domains	only source	ADA	ADA SDM	ADA Union	IADA	IADA SDM
0.9	99.31	-	-	-	99.61	99.52
0.8	99.20	-	-	-	99.53	99.36
0.7	98.40	-	-	-	99.20	99.01
0.6	93.51	-	-	-	95.68	95.11
0.5	84.11	87.10	86.83	87.62	89.90	89.51

TABLE I: Target classifier accuracy on incrementally transformed MNIST dataset. The last row represents the final accuracy on the maximally transformed input samples. While the naive alignment to the union of all target datasets already improves in performance of an approach with only access to the final domain, IADA results in further significant accuracy improvement. SDM only slightly affects the target performance and the combination IADA SDM continues to outperform the original ADA baseline.

Table I shows the target domain classification accuracy of 1-step adaptation methods against their incremental counterparts which continue optimising the target encoder across domains with incrementally increasing domain shift. Furthermore, we test the methods in combination with GAN-based Source Domain Modelling (SDM) introduced in Section III-A.

As displayed in Table I, all domain adaptation approaches outperform pure source-optimised models, while incremental

domain adaptation provides additional benefits over regular ADA. Utilising the union of all target domains as target domain for regular ADA increases model accuracy in the final target domain only slightly above using only one target domain and still performs worse than IADA. Finally, the SDM variants of all approaches only result in minor performance reductions, while reducing memory requirements significantly.

To investigate classification accuracy in dependence of the number of available intermediate domains, the MNIST digits are rescaled further to 0.3 of the original height and evaluated with varying numbers of equally spread intermediate domains with IADA and IADA SDM. We chose to increase the deformation in comparison to the earlier experiment to increase the complexity of the domain adaptation task and enable evaluating a wider range of sub-domain discretisations.

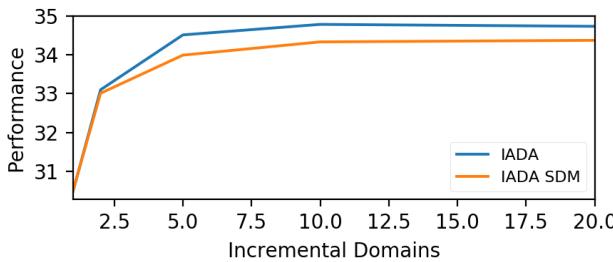


Fig. 5: Classifier accuracy of IADA in final target domain with varying number of intermediate domains for horizontal compression of 0.3. The strong digit deformation leads to a challenge for domain adaptation. Results show the benefits of separating large domain shifts into incremental domain adaptation steps for IADA. Maximal performance for this adaptation scenario is achieved between 10 and 20 incremental domains and further increase does not significantly influence the final target accuracy.

Separating larger shifts into incremental steps as displayed in Figure 5 enables us to address the problem with a curriculum of easier tasks. Above a certain threshold however target performance remains consistent with further increase of the number of target domains. For the domain adaptation from MNIST to its rescaled copy, the benefits of incremental domain adaptation saturate at around 10 to 20 intermediate domains. However, more complex transformations can rely on even more incremental approaches.

B. Free Space Segmentation

An area of active research in autonomous driving is the detection of traversable terrain. Especially when utilising images as input, methods often rely on collecting data in all possible deployment domains and weather conditions.

We evaluate IADA in this context for a drivable-path segmentation method based on segments of the Oxford RobotCar dataset [12]. The employed path segmentation labels are generated in a self-supervised setting based on [26]. The dataset consists of approximately hour-long driving sessions from different days collected over the course of a



Fig. 6: Incremental changes of lighting conditions in the Oxford10k dataset from early morning (top row) to late night (bottom row).

year. Based on the nature of the dataset, we approximate the scenario of continuous application by picking five datasets to represent different daylight conditions from morning to evening and train on a labelled source dataset based on morning data as seen in Figure 6.

The resulting 5 intermediate domains were chosen to represent incremental change in lighting conditions and serve as a proxy for the online deployment scenario. Each domain consists of about 2000 images, rescaled for the evaluation to a size of 128 by 320 pixels. Pixel-wise segmentation labels for training are available only for the source domain, while the approach utilises test labels for the evaluation in all domains.

For all segmentation tasks, we employ an adaptation of the ENet [27] architecture which presents a compromise of model performance and size. The architecture focuses on strong segmentation accuracy as well as reasonable computational requirements, which makes it a strong contender for online deployment on mobile platforms. For the discriminator, we split the ENet architecture just before the upsampling stages (see [27]) and employ an additional 4-layer convolutional discriminator. Similar to Section IV-A we duplicate all parameters before the architecture split to be utilised as source and target encoders.

The results for drivable-path segmentation are represented in Table II. Similar to the application on the synthetic domain shift dataset in Section IV-A, IADA outperforms one-step domain adaptation. ADA with respect to the union over all incremental target domains is more accurate than with only the final domain but not as exact as IADA. Again, SDM slightly reduces the target performances, however rendering the storage of significant source datasets unnecessary.

In real-world scenarios, we cannot ensure smoothness over the appearance changes and the turning-on of street lights for the final target domain indeed represents a step change in our environment. It is to be expected that more continuous domain shifts would increase the advantages of IADA as displayed in the context of synthetic data in Section IV-A.

In comparison to the toy example, the combination of all target domains only leads to minor improvement over the regular application of ADA with only the final target domain. The final target domain's instantaneous change in lighting due to the switching-on of the street lights leads to significant

target domains	only source	ADA	ADA SDM	ADA Union	IADA	IADA SDM
morning	91.62	-	-	-	91.60	91.77
midday	90.70	-	-	-	91.05	90.50
afternoon	89.10	-	-	-	89.91	89.53
evening	87.08	-	-	-	89.01	87.34
night	76.27	78.67	77.12	78.83	80.21	79.37

TABLE II: Mean average precision results for segmentation task in continuous deployment scenario. Applying domain adaptation with respect to the union of all target domains slightly increases performance. The incremental adaptation approach leads to further improvement, while the approximation of the source domain only slightly reduces performance.

differences to previous target domains. This scenario renders domain adaptation to the union over all target domains less efficient and emphasising the importance of an incremental method, which focuses on the current target domain.

With computation times of approximately 26 minutes for the adaptation to a new incremental target domain on an NVIDIA GeForce GTX Titan Xp GPU, we can potentially deploy the system on vehicles to adapt to the currently encountered domain at a rate of about 55 times a day in continuous deployment. The extension with source domain modelling reaches computation times of 29 minutes resulting in nearly 50 updates per day.

V. DISCUSSION

While IADA’s principal benefits are based on continuous access to the incremental shifts between source and target domains, the evaluation for drivable-path segmentation with our offline datasets builds on a sequence of distinct target domains extracted from the Oxford RobotCar Dataset. The approach can be extended easily to more continuous alignment to the online perceived data domain via the utilisation of sliding window sampling during deployment. Interestingly, it was shown in Section IV-A that the benefits of dividing the target domains further for IADA can saturate when the intermediate domains are becoming increasingly similar.

IADA relies on access to the incremental shifts in the appearance of our environment. With limited access or step-wise changes in the perceived environment the approach degrades to regular adversarial domain adaptation. In particular, this paradigm becomes visible in our segmentation datasets where the turning-on of the streetlights leads to an instantaneous change in the appearance of the environment.

However, this instantaneous domain shift caused by the final domain’s lighting change further emphasises the benefits of an incremental approach over simply using the union over all target domains for regular domain adaptation. IADA significantly outperforms this method as it specifically optimises for the currently relevant target domain.

All experimental results noted in our work are based on the confusion loss for domain adaptation [5]. An adaptation of the Wasserstein GAN framework [28] for domain adaptation leads

to (on average) slightly more stable training and statistically insignificantly improvement in performance. However, we focused on the confusion loss formulation as, due to the additional critic training rounds required for the WGAN framework, it leads to significantly lower training duration.

While increased computational effort might not be critical for server-side computation, it can limit applications of embedded systems. In the context of cloud computing or larger platforms with significant data storage volumes, the minor accuracy loss can be prevented when applying the original formulation for IADA in Section III.

VI. CONCLUSION AND FUTURE WORK

We present a method for addressing the task of domain adaptation in an incremental fashion, adapting to a continuous stream of changing target domains. Furthermore, we introduce an approach for source domain modelling, training a GAN to approximate the feature distribution in the source domain to render the domain adaptation step independent of retaining large amounts of source data. Both methods are evaluated first on synthetically shifted versions of rescaled MNIST digits for illustration purposes and full access to the number of intermediate domains. Furthermore, we empirically demonstrate their performance on the real-world task of drivable-path segmentation in the context of autonomous driving.

The field of continual training during deployment provides many possible benefits as models can be adapted to the currently encountered environment and learn from data unavailable during offline training. However, the approach also opens up new security challenges. The well-known problem of perpetrators introducing adversarial samples to the system could lead to not only corruption of the current prediction but prolonged distortion of the model. This area represents an essential direction for further research on defending against adversarial examples. Further indispensable extensions of this work include addressing the additional problem of catastrophic forgetting in lifelong-learning scenarios. This direction has the potential to further reduce computational requirements as it will discard the necessity to readapt to once encountered target domains.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the UKs Engineering and Physical Sciences Research Council (EPSRC) through the Programme Grant EP/M019918/1 and the Doctoral Training Award (DTA) as well as the support of the Hans-Lenze-Foundation. Additionally, the donation from NVIDIA of the Titan Xp GPU used in this work is gratefully acknowledged.

REFERENCES

- [1] Sivalogeswaran Ratnasingam and Steve Collins. Study of the photodetector characteristics of a camera for color constancy in natural scenes. *JOSA A*, 27(2):286–294, 2010.
- [2] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. *CoRR*, abs/1605.06636, 2016.

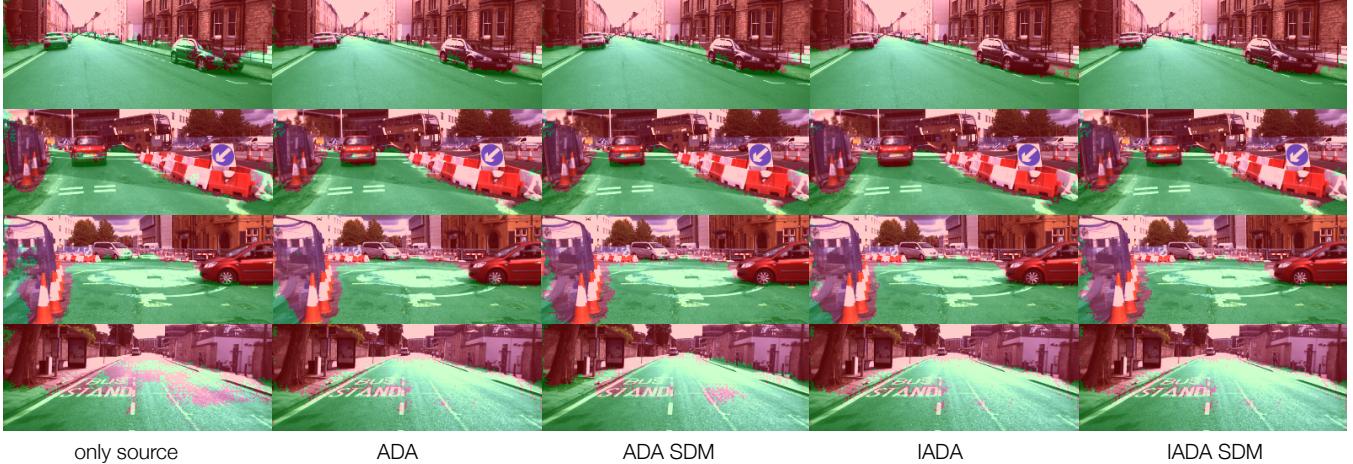


Fig. 7: Segmentation predictions for the final target domain overlaid on the input images (green and red represent traversable terrain and obstacles respectively). From left to right: training only on the source domain, ADA, IADA, ADA SDM, IADA SDM. Adversarial domain adaptation consistently outperforms source training, with IADA providing additional benefits. When combined with SDM both approaches result in only slightly lower accuracy. While only source domain trained models display obvious weaknesses correlated to the different street illumination, the main benefits of IADA against ADA can be found in details such as more distinct obstacle boundaries and less noisy segmentation. The slight performance reduction based on SDM is qualitatively negligible and mostly visible in the quantitative evaluation in Table II

- [3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Victor Lempitsky, Urun Dogan, Marius Kloft, Francesco Orabona, and Tatiana Tommasi. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17:1–35, 2016.
- [4] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.
- [5] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2017.
- [6] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016.
- [7] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Xingchao Peng, Sergey Levine, Kate Saenko, and Trevor Darrell. Towards adapting deep visuomotor representations from simulated to real environments. *arXiv preprint arXiv:1511.07111*, 2015.
- [8] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.
- [9] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcn in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.
- [10] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. *CoRR*, abs/1612.07828, 2016.
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [12] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.
- [13] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.
- [14] Winston Churchill and Paul Newman. Experience-based Navigation for Long-term Localisation. *The International Journal of Robotics Research (IJRR)*, 2013.
- [15] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 198–203. IEEE, 2013.
- [16] Sungeun Hong, Woobin Im, Jongbin Ryu, and Hyun S Yang. Ssppdan: Deep domain adaptation network for face recognition with single sample per person. *arXiv preprint arXiv:1702.04069*, 2017.
- [17] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *arXiv preprint arXiv:1603.06432*, 2016.
- [18] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. *CoRR*, abs/1612.01939, 2016.
- [19] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.
- [20] Aniruddha Kembhavi, Behjat Siddique, Roland Miezianko, Scott McCloskey, and Larry S Davis. Incremental multiple kernel learning for object recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 638–645. IEEE, 2009.
- [21] Vídit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 577–584. IEEE, 2011.
- [22] Alex Bewley, Vitor Guizilini, Fabio Ramos, and Ben Upcroft. Online self-supervised multi-instance segmentation of dynamic objects. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1296–1303. IEEE, 2014.
- [23] Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 867–874, 2014.
- [24] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.
- [25] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- [26] Dan Barnes, William P. Maddern, and Ingmar Posner. Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. *CoRR*, abs/1610.01238, 2016.
- [27] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.
- [28] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.