IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 1, March 2023, pp. 96∼105
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i1.pp96-105

Boosting auxiliary task guidance: a probabilistic approach

Irfan Mohammad Al Hasib[1], Sumaiya Saima Sultana[1], Imrad Zulkar Nyeen[2], Muhammad Abdus Sabur[2]
[1]Department of Mechanical Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
[2]Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

Article Info

Article history:

Received Dec 26, 2021
Revised Aug 31, 2022
Accepted Sep 26, 2022

Keywords:

Auxiliary task guidance
Computer vision
Deep learning
Multi task learning
Visual odometry

ABSTRACT

This work aims to introduce a novel approach for auxiliary task guidance (ATG). In this approach, our goal is to achieve effective guidance from a suitable auxiliary task by utilizing the uncertainty in calculated gradients for a mini-batch of samples. Our method calculates a probabilistic fitness factor of the auxiliary task gradient for each of the shared weights to guide the main task at every training step of mini-batch gradient descent. We have shown that this proposed factor incorporates task specific confidence of learning to manipulate ATG in an effective manner. For studying the potency of the method, monocular visual odometry (VO) has been chosen as an application. Substantial experiments have been done on the KITTI VO dataset for solving monocular VO with a simple convolutional neural network (CNN) architecture. Corresponding results show that our ATG method significantly boosts the performance of supervised learning for VO. It also out performs state-of-the-art (SOTA) auxiliary guided methods we applied for VO. The proposed method is able to achieve decent scores (in some cases competitive)compared to existing SOTA supervised monocular VO algorithms, while keeping an exceptionally low parameter space in supervised regime.

Corresponding Author:

Irfan Mohammad Al Hasib
Department of Mechanical Engineering, Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
Email: irfanhasib.me@gmail.com

1. INTRODUCTION

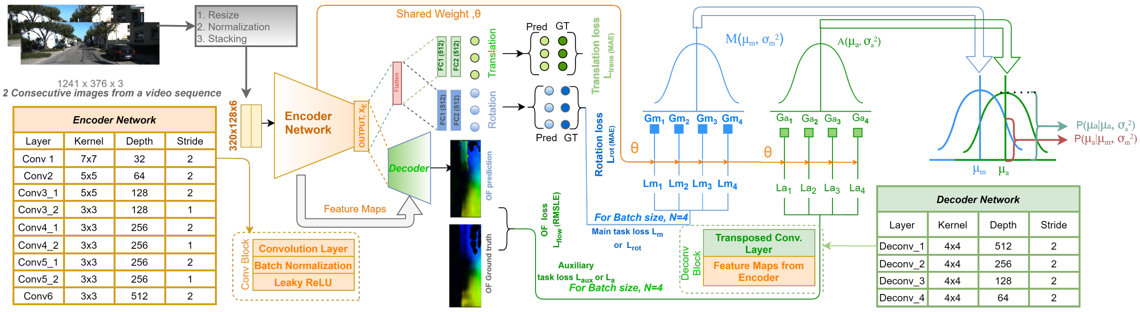
Recent research shows that various forms of multi-task learning (MTL) are being used to boost the performance of neural networks (NN) beyond their capacity [1]. The goal of MTL is to maximize the performance of several tasks by learning multiple tasks together while auxiliary task learning is directly concerned with better performance of the primary task [2]. In this work, we present an effective approach to gain guidance from auxiliary tasks. The novelty of the proposed approach is that it can effectively control the contribution of auxiliary task gradients for each shared weight by measuring its suitability for fitting into the main task's loss gradient distribution. For investigating the effectiveness of the proposed method, we have chosen the complex problem of monocular visual odometry (VO) pose estimation. Geometric approaches of VO [3]–[6] work very well for known environments, but require consistency in camera calibrations [7]. However, learning based approaches show superiority in robustness to inconsistent environment[8]. Complex architectures of deep learning (DL) solutions capture the high complexity of the VO problem. However, DL based approaches possess limitations like higher inference time, larger memory requirements, and overfitting tendencies. Simpler architectures may create balance between these challenges. But merely using a simple architecture is not good enough for solving complex VO problems [9]. Performance boosting techniques like MTL, auxiliary task learning (ATL) can be a solution here. Costante and Ciarfuglia [10], Yang *et al.* [11] are examples where MTL approaches have been embraced in pose estimation. Using the proposed ATG method, we solve them monocular VO pose estimation successfully problem with relatively simple architecture.

Developing better guidance methods for MTL and ATL is a primary research question. Chen *et al.* [12] performs normalization of gradients to balance learning between multiple tasks. Yu *et al.* [13] manipulates directions of the gradients to provide better guidance. Du *et al.* [14] quantifies similarity between this by measuring the cosine similarity between gradient vectors of two tasks and therefore tuning for a suitable threshold for similarity value. Our proposed approach measures similarity in a more precise manner by considering each shared weight gradient separately. Unlike existing approaches, we weigh the similarity with task specific confidence of learning as well. For the chosen field of application of VO, traditional geometric methods produced state-of-the-art solutions for pose estimation including [6], [5], but they are prone to motion drift. Supervised learning-based methods solve this challenge because they are more robust to unstable environments [15]. However, they possess an additional challenge of requiring complex architectures or having a huge number of hyper-parameters [16]–[21]. Due to these conflicts, ultimately most recent works are focusing on unsupervised learning and being successful with much higher margin [11], [22], [23]. Among MTL based supervised approaches for VO problem, latent space VO (LS-VO) [10] learns a low dimensional optical flow (OF) subspace with pose estimation jointly but still works in a huge parameter space. Our proposed ATG approach helps enable a supervised learning method to perform well even in a much lower parameter space than existing methods for the complex problem of VO. At a glance, the contributions of this paper are: i) Proposing a new approach for providing effective and better guidance from auxiliary task. ii) Demonstrating the effectiveness of the proposed method by solving the monocular VO problem using a tiny network compared to existing supervised models with OF subspace learning as auxiliary task.

2. METHODOLOGY

2.1. NN architecture specification

The complete architecture, illustrated in Figure 1 can be divided into two major sections, encoder section and task specific section. The encoder section is a modified FlowNet architecture [24], reducing its depth by half for every layer. This section is the feature extractor of the framework and is shared by both tasks. The task specific section, consisting of three parts, is dedicated for translation estimation, rotation estimation and flow image prediction. Rotation and translation estimation parts of the network are based on separate sequences of dense fully connected layers and a decoder network estimates OF.



Encoder Network			
Layer	Kernel	Depth	Stride
Conv 1	7x7	32	2
Conv2	5x5	64	2
Conv3_1	5x5	128	2
Conv3_2	3x3	128	1
Conv4_1	3x3	256	2
Conv4_2	3x3	256	1
Conv5_1	3x3	256	2
Conv5_2	3x3	256	1
Conv6	3x3	512	2

Decoder Network			
Layer	Kernel	Depth	Stride
Deconv_1	4x4	512	2
Deconv_2	4x4	256	2
Deconv_3	4x4	128	2
Deconv_4	4x4	64	2

Figure 1. Visual representation of the architecture and algorithm

We scaled down the input images from $(1241, 376, 3)$ to $(320, 128, 3)$ which reduces parameters and helps to overcome overfitting without compromising result accuracy. Images were normalized with mean centered to 0 ranging from -0.5 to +0.5. The network takes two consecutive images from each particular sequence of KITTI VO dataset and stacks them depth-wise as input. In Figure 1, output $\mathbf{X_E}$ is generated after the image passes through the 9 shared layers of the encoder. Each shared layer block consists of 2D-convolution layer, batch normalization and leaky rectified linear unit (leaky ReLU). The flow image prediction section uses output $\mathbf{X_E}$ directly as the input. Flattened $\mathbf{X_E}$ is used as the input of translation estimation and rotation estimation sections, both consisting of dense layers. Final outputs of the entire framework are 6 degrees of freedom (DOF) pose estimation and flow image predictions.

2.2. Probabilistic auxiliary task guidance

As shown in Figure 1, the network learns three tasks with three different loss functions-translation loss(L_{trans}), rotation loss(L_{rot}) (as main task loss), OF subspace learning loss(L_{flow}) (as auxiliary task loss). In ATG learning, the total loss is usually defined as:

$$L_{total}(\theta, \phi_{main}, \phi_{aux}) = \beta_m L_{main}(\theta, \phi_{main}) + \beta_a L_{aux}(\theta, \phi_{aux})$$

$$\text{weight } \theta \text{ can be updated as, } \theta_{new} = \theta + \alpha(\beta_m \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \tfrac{dL_{main}}{d\theta} + \beta_a \tfrac{1}{N} \sum_{i=1}^{N} \tfrac{dL_{aux}}{d\theta})$$

Here, α= learning rate, β_m= main task loss coefficient, β_a= auxiliary task loss coefficient, θ= weights of shared layer, ϕ_{main}= weights of task specific layers for main task, ϕ_{aux}= weights of task specific layers for auxiliary task, N= mini-batch size. One of the key research questions in ATL is to choose an optimum coefficient (β_a) to encourage positive transfer and blocking negative transfer from an appropriate auxiliary task [1], [14]. Our approach finds a solution to this question by tuning β_a initially and then optimizing it extensively with a probabilistic factor calculated for each shared weight that prioritizes assistance from the auxiliary task with respect to its guiding capability. In this section, we present the approach of calculating this factor and discuss how it allows us to integrate both task specific confidence of learning and task similarity in the guidance process.

From central limit theorem, we can say that the gradients of a mini-batch belong to a certain normal distribution. Let the mean of the gradients for main task, $\frac{1}{N} \sum_{i=1}^{N} \frac{dL_{main}}{d\theta}$ be μ_m and the mean of the gradients for auxiliary task be μ_a. The distributions of gradients for the main and auxiliary task are denoted as $M(\mu_m, \sigma_m^2)$ and $A(\mu_a, \sigma_a^2)$ respectively. Calculated probabilities of μ_a in both distributions have been expressed as $P(\mu_a|\mu_m, \sigma_m^2)$ and $P(\mu_a|\mu_a, \sigma_a^2)$. This calculated $P(\mu_a|\mu_m, \sigma_m^2)$ indicates what probability would μ_a have if it belonged to the distribution of the main task M. In other words, it signifies how much μ_a fits the current distribution M as a random numeric value. Hence, it could be a reasonable parameter to decide how much auxiliary task loss should contribute to the total loss. It can be incorporated by using it as a multiplication factor with auxiliary task loss coefficient. But empirical values of gradients and their variances reveal that the probability $P(\mu_a|\mu_m, \sigma_m^2)$ values vary in between a very wide range($10^1 \sim 10^4$), shown in Figure 2. Hence, using probability $P(\mu_a|\mu_m, \sigma_m^2)$ merely as the multiplication factor makes the gradients unstable by changing them drastically. Two types of scaling are done to handle this issue. In the first method, we divide these probabilities by their maximum value in respective layer's weights. Thus it is restricted between (0,1). This method is named probabilistic factor (PF) method.

$$\Delta\theta = \beta_m \mu_m + [P(\mu_a|\mu_m, \sigma_m^2)/P_{max}]\beta_a \mu_a \tag{1}$$

Where, P_{max} = maximum value of probability in respective layers. However, probability values with small magnitude are at risk of getting vanished especially when variance values are high. Taking log of probability does not help much in this issue. So,in the second method, we propose to scale the probability $P(\mu_a|\mu_m, \sigma_m^2)$ by dividing it with $P(\mu_a|\mu_a, \sigma_a^2)$ which is the probability of the same variable μ_a in its own distribution. We denote this method as the probability ratio factor (PRF) (1) method. It has performed best in our experiments for reasons we will explain gradually in later sections. We have defined the ratio of two probabilities as relative probability factor, $\rho(m, a)$ and updated (1) as (3).

$$\rho(m, a) = \frac{P(\mu_a|\mu_m, \sigma_m^2)}{P(\mu_a|\mu_a, \sigma_a^2)} \tag{2}$$

$$\Delta\theta = \beta_m \mu_m + \rho(m, a)\beta_a \mu_a \tag{3}$$

$$\text{So, } \theta_{new} = \theta + \alpha(\beta_m \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \tfrac{dL_{main}}{d\theta} + \rho(m, a)\beta_a \tfrac{1}{N} \sum_{i=1}^{N} \tfrac{dL_{aux}}{d\theta})$$

The value of $\rho(m, a)$ is constrained within a suitable range (shown in Figure 2(a), 2(b), and 2(c)). So the ratio does not change drastically for consecutive training steps, the learning process becomes more stable. For the purpose of analysis ,we have introduced two novel terms: task confidence, ζ and task similarity, τ as:

$$P(\mu_a|\mu_m, \sigma_m^2) = \frac{1}{\sigma_m\sqrt{2\pi}} \exp(-\tfrac{1}{2}(\tfrac{\mu_a - \mu_m}{\sigma_m})^2); \zeta(m) = \frac{1}{\sigma_m\sqrt{2\pi}}; \tau(m, a) = \exp(-\tfrac{1}{2}(\tfrac{\mu_a - \mu_m}{\sigma_m})^2)$$

The term $\frac{1}{\sigma_m\sqrt{2\pi}}$ is inversely proportional to standard deviation (σ_m) of the distribution. For a particular mini-batch, lower variance of the gradients will indicate stable learning. Intuitively, we interpret it as a measure of confidence of this distribution M.The term $(\frac{\mu_a-\mu_m}{\sigma_m})$ is the measure of distance between μ_a and μ_m for unit σ_m. So, mathematically, $\tau(m,a)$ will measure similarity(not distance) very strictly if the distribution has low variance and vice versa. For auxiliary task, $\tau(a,a) = \exp(-\frac{1}{2}(\frac{\mu_a-\mu_a}{\sigma_a})^2) = 1$. So, from (2), probability ratio,

$$\rho(m,a) = \frac{P(\mu_a|\mu_m,\sigma_m^2)}{P(\mu_a|\mu_a,\sigma_a^2)} = \frac{\zeta(m)\tau(m,a)}{\zeta(a)\tau(a,a)} = \frac{\zeta(m)}{\zeta(a)}\tau(m,a) \tag{4}$$

Here, we have defined $\zeta(m)/\zeta(a)$ as the relative task confidence of the main task compared to the auxiliary task which helps main task in different scenarios, summarized in Table 1. If the main task is learning with relatively higher confidence and auxiliary task gradient still has a good similarity (despite the high confidence value of the main task) that is the most desired scenario for ATG. Thus our approach ensures efficient and effective guidance from the auxiliary task by avoiding negative transfer in critical scenarios.

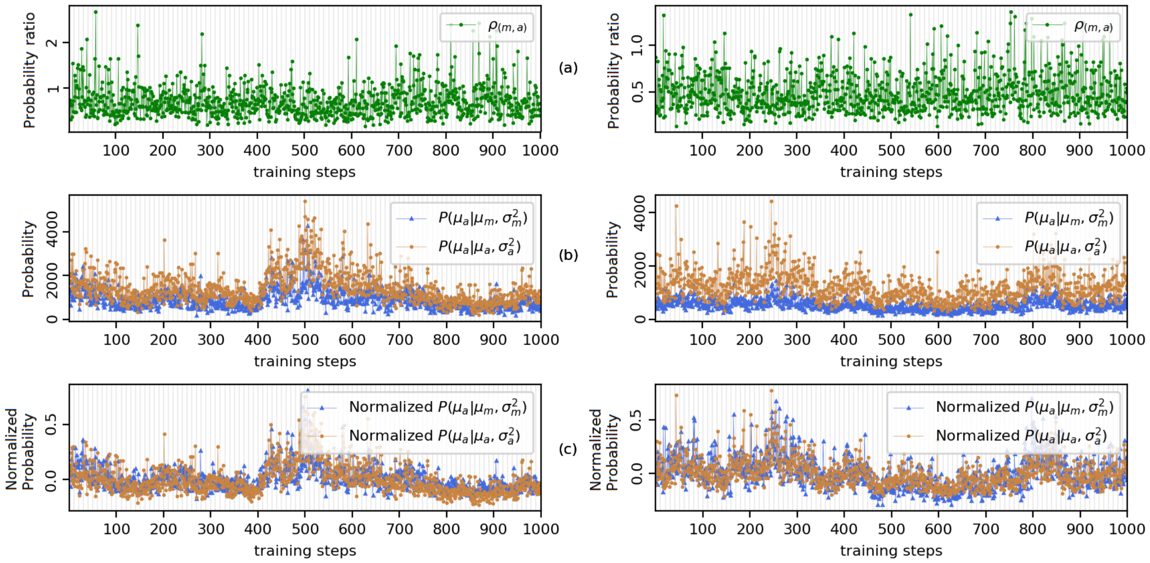


Figure 2. Probability ratio, raw probability and normalized probabilities for main task and auxiliary task. Data is collected for one random weight of layer 1 for 1,000 consecutive training steps. Left plot is for 15^{th} epoch and right plot is for 30^{th} epoch among 30 epochs. a) probability ratio b) raw probabilities and c) normalized probabilities.

Table 1. Possible scenarios of the PRF method

Scenario	$\zeta(m)/\zeta(a)$	$\tau(m,a)$	Auxiliary task guidance
Case 1	high	low	moderate guidance
Case 2	high	high	helps positive guidance
Case 3	low	low	blocks negative guidance
Case 4	low	high	moderate guidance

From the above definition of the given terms, we can write, $P(\mu_a|\mu_m,\sigma_m^2) = \zeta(m)\tau(m,a)$. Now, probabilities vary within a wider range around $(10^1 \sim 10^4)$ [2(b)]. Since task similarity, $\tau(m,a)$ lies between $(0 \sim 1)$, so the value of ζ effectively controls the range of probability values. Consequently correlation between $\zeta(m)$ and $\zeta(a)$ shown in Figure 3 causes correlation between the probabilities demonstrated in Figure 2(c). That's why the scaling effectively keeps the probability ratio $\rho(m,a)$ within a suitable range [Figure 2(a)]. The relative probability factor reduces the effect of common sources of variance among the distributions. Thus it emphasizes on the task specific variance that gives information only about relative performance of the tasks. It will be explained in detail in 2.4.. Also, auxiliary task gradients are more stable compared to the main task, (discussed in 2.3.). So, the relative probability ($\rho(m, a)$) gives us a better estimate of relative performance of the main task compared to auxiliary task. Note that the parameter β_a is a constant coefficient (tuned as a hyperparameter) for all the weights but $\rho(m, a)$ is calculated separately for each of the weights at every mini-batch gradients update.

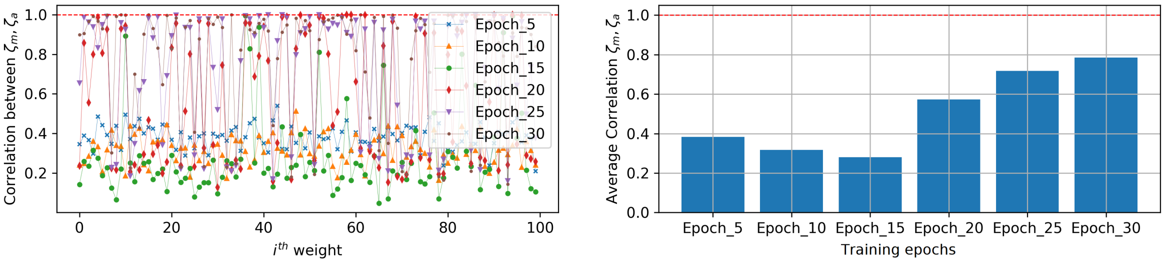


Figure 3. Correlation of $\zeta(a)$ and $\zeta(m)$ for i=100 different weights (θ_i) (X-axis). Here, $\zeta(m)$ and $\zeta(a)$ is collected from 1,000 consecutive steps of a single epoch and $corr(\zeta(m), \zeta(a))$ for each of the i^{th} weight is measured from this 1,000 confidence pairs. To comprehend the information, right side plots are numerical average for every epochs of the left plots

2.3. Optical flow subspace learning as auxiliary task

Estimating a lower dimensional representation of the dense OF field from one pair of raw red green blue (RGB) images is the auxiliary task learned by our network. The chosen auxiliary task is easier to learn than the main task. This is because latent OF representation estimation loss is measured from the entire OF image where the odometry loss is measured from 6 sparse pose values. Also we are learning OF subspace by minimizing pixel-wise root mean squared log error (RMSLE) (while precise raw OF learning requires using root mean squared error (RMSE) loss). This makes the learning task even easier [10]. Finally, the capability of OF task for helping pose estimation can be proved from vanilla MTL results from experiment with ATG method in Table 2.

Table 2. Comparative results of different approaches

Sequences	NN		ATG		PF		PRF	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
04	13.5811	2.0449	12.0222	2.2133	11.7648	0.9098	15.3180	1.1278
05	10.9609	2.1660	11.9858	2.1768	9.4700	1.8930	6.9010	1.4982
07	13.1821	4.5040	11.6105	4.3484	8.0245	3.7180	6.9114	2.5165
10	21.9880	8.0013	19.3150	3.9233	13.5378	4.2081	11.6025	3.4710

t_{rel}: mean translational RMSE drift (%) on length of 100m-800m.
r_{rel}: mean rotational RMSE drift (deg/100m) on length of 100m-800m.

2.4. Sources of variance in gradients

In this section, we have analyzed the sources of variances in the task specific loss gradients calculated with respect to the shared weights. The loss gradients can be expressed in (5) using chain rule of differentiation:

$$G_t = \frac{dl_t}{d\theta} = \frac{dz}{d\theta} \cdot \frac{dA}{dz} \cdot \frac{dl_t}{dA} \tag{5}$$

Where, $z = \theta x + b$, $A = f(z)$, x = input coming from previous layer, l_t = function of loss for task t, θ = shared weight; b = bias; f:activation function. Since the shared layers are convolutional layers the above equations element should be corresponding matrices like $\mathbf{Z} = \Theta\, \mathbf{x} + \mathbf{b}$. But we have considered scaler variables for simplicity. From (5), the gradient equations of both tasks will be the same except $\frac{dl_m}{dA}$ for main task and $\frac{dl_a}{dA}$ for auxiliary task. So we can write $dz/d\theta \times dA/dz = F(x)$ since z and A both are functions of x and $dl_t/dA = H(l_t)$. Consequently, the common term $F(x)$ is responsible for the correlation in the variances of these gradients as shown in Figure 3. Generalizing the equation for gradient calculation:

$$G_t = \frac{dl_t}{d\theta} = F(x)H(l_t) \tag{6}$$

We propose that incorporating relative task confidence $\zeta(m)/\zeta(a)$, allows us to diminish the effect of the part of variance coming from the common source. So the only functional part of the variance is coming from task specific losses. This claim can be proved by following example:
Let X = $\{x_1, x_2, ..., x_D\}$; L_m = $\{l_1^m, l_2^m, ..., l_D^m\}$; $L_a = \{l_1^a, l_2^a, ..., l_D^a\}$ (D is the size of dataset) be a set of inputs and corresponding main task and auxiliary task losses. Let's consider a mini batch of n samples, X_b = $\{x_i | i \in [1, n], i \in \mathbb{N}\}$ where $X_b \subseteq X$; L_b^m = $\{l_i^m | i \in [1, n], i \in \mathbb{N}\}$ where $L_b^m \subseteq L^m$; L_b^a = $\{l_i^a | i \in [1, n]\}$ where $L_b^a \subseteq L_a$. Let, $n = 3$. For ease of expressing relations, elements of set $F(X_b)$ are denoted by a, b, c respectively. Similarly, $H(L_b^m)$ = $\{u, v, w\}$ and $H(L_b^a)$ = $\{p, q, r\}$. So, from (6), gradients can be expressed as:

$$G_1^m = au; G_2^m = bv; G_3^m = cw; G_1^a = ap; G_2^a = bq; G_3^a = cr$$

$$\tfrac{\zeta(m)}{\zeta(a)} = \frac{\sqrt{E({G_b^a}^2) - E(G_b^a)^2}}{\sqrt{E({G_b^m}^2) - E(G_b^m)^2}} = \frac{\sqrt{[N({G_1^a}^2 + {G_2^a}^2 + {G_3^a}^2) - (G_1^a + G_2^a + G_3^a)^2]}}{\sqrt{[N({G_1^m}^2 + {G_2^m}^2 + {G_3^m}^2) - (G_1^m + G_2^m + G_3^m)^2]}}$$

$$\frac{\zeta(m)}{\zeta(a)} = \frac{\sqrt{[N(a^2p^2 + b^2q^2 + c^2r^2) - (ap + bq + cr)^2]}}{\sqrt{[N(a^2u^2 + b^2v^2 + c^2w^2) - (au + bv + cw)^2]}} \tag{7}$$

If the loss dependent variable sets (p, q, r in numerator and u, v, w in denominator) have very low variance compared to the input dependent variable set a, b, c; then the change in values of σ_a and σ_m will be dominated by mostly a, b, c. In (7), the denominator and the numerator both contain input dependent variables a, b, c which consequently causes correlation between $\zeta(m)$ and $\zeta(a)$ (observed in Figure 3). The lower the variance of p, q, r and u, v, w will be (compared to the variance of a, b, c), the higher correlation will be eventually. Let, σ_{abc} = standard deviation of the input dependent variables, σ_{uvw} = standard deviation of main task loss dependent variables, σ_{pqr} = standard deviation of auxiliary task loss dependent variables. Let's discuss the effect of σ_{abc} and σ_{uvw} in relative task confidence, $\zeta(m)/\zeta(a)$, considering σ_{pqr} remains almost constant. Confidence of main task $\zeta(m)$ decreases when σ_m increases. This can happen in 3 possible cases: *Case 1*: only σ_{abc} increases- In this case the numerator and denominator both will increase resulting comparatively no notable change in relative task confidence factor,$\zeta(m)/\zeta(a)$. So it diminishes the effect of high variance of main task loss gradient if the variance is being caused by input dependent sources (common source). *Case 2*: σ_{uvw} increases- In this case only the denominator will increase, resulting in significantly lower $\zeta(m)/\zeta(a)$ value. So here $\zeta(m)/\zeta(a)$ is considering the effect of high variance of main task loss gradient significantly only when the variance is being caused by task specific sources (uncommon source which is task loss dependent). *Case 3*: Both σ_{abc} and σ_{uvw} increases- In this case the numerator will increase but the denominator will increase more since it is function of both a, b, c and u, v, w, resulting moderately lower value of $\zeta(m)/\zeta(a)$ (not as low as case 2). So, $\zeta(m)/\zeta(a)$ seems to decrease the effect of high variance of main task loss gradient moderately because the variance is being caused by both task loss specific sources and input dependent sources. Above 3 cases demonstrate, how relative task confidence is less affected by common source of variances.

Algorithm 1 Algorithm (PRF)

```
$Init\ \theta, \phi_{main}, \phi_{aux}$
$set\ \alpha, \beta_m, \beta_a$
while $epoch$ do
    for $mini\text{-}batch\ in$ Dataset do
        $G_i^m \leftarrow L_{main}^i; G_i^a \leftarrow L_{aux}^i\ \forall i; i = [1, N] \land i \in \mathbb{N}$
        $\mu_m \leftarrow \frac{1}{N}\sum_{i=1}^{N} G_i^m, \sigma_m^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N}(G_i^m - \mu_m)^2$
        $\mu_a \leftarrow \frac{1}{N}\sum_{i=1}^{N} G_i^a, \sigma_a^2 \leftarrow \frac{1}{N}\sum_{i=1}^{N}(G_i^a - \mu_a)^2$
        $calculate\ \zeta_m, \zeta_a, \tau(m, a)\ from\ \mu_m, \sigma_m, \mu_a, \sigma_a$
        $\rho(m, a) \leftarrow (\zeta_m/\zeta_a) \times \tau(m, a)$
        $\Delta\theta \leftarrow \beta_m\mu_m + \rho(m, a)\beta_a\mu_a$
        $\theta \leftarrow \theta + \alpha\Delta\theta$
        $\phi_{main} \leftarrow \phi_{main} + \alpha\beta_m\mu_m$
        $\phi_{aux} \leftarrow \phi_{aux} + \alpha\beta_a\mu_a$
    end for
end while
```

3. EXPERIMENTAL RESULTS

KITTI VO dataset [25] is used to learn pose estimation. This dataset contains 11 sequences; seq. 0-3, 6, 8-9 have been used for training while 4, 5, 7 and 10 have been used for validation. However, KITTI VO dataset does not include OF images for these sequences. We have trained FlowNetS [24] architecture separately using KITTI flow dataset and used the trained FlowNet to generate OF ground truth images for VO dataset. OF templates are trained using root mean squared logarithmic error (RMSLE) [10] to learn the OF subspace effectively (Figure 4), where 4(a) and 4(b) is consecutive image pair, 4(c) ground truth, 4(d) predicted OF. For model training, We have used (320 × 128) images with batch size 16. After hyperparameter tuning, we have found the best fitted model with learning rate 0.0005, $\beta_t = 1$, $\beta_r = 10$ and $\beta_a = 0.1$, gradient clipping has been applied to prevent overfitting. TensorFlow and Keras DL framework have been used for all experiments with machine specifications: Intel Core i7-9750H CPU@2.60GHz (12 CPUs), 16 GB RAM and NVIDIA GeForce GTX 1660Ti GPU.

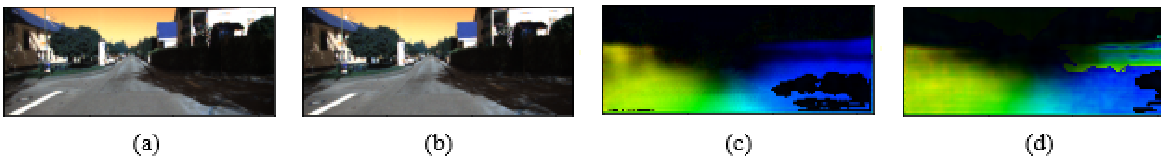


Figure 4. Optical flow subspace estimation (a),(b) image pair (c), ground truth, and (d) prediction

Figure 5 demonstrates (from left to right): i) simple NN method (without ATG), ii) vanilla ATG method, iii) PF method, iv) PRF method, v) Library for Visual Odometry 2 - Monocular (LIBVISO2-M) method, vi) Oriented FAST and rotated BRIEF-simultaneous localization and mapping (ORB-SLAM2) method. Here, v) [6] and vi) [26] are popular geometric methods commonly used for solving VO problem.They are given to compare PF and PRF methods' performance with respect to existing geometric methods. From i) to iv), it is evident that both PF and PRF method boosts the performance of simple NN as well as vanilla ATG method with a good margin (Figure 5, Table 2). Thus the claim regarding parameter reduction with our method is justified. It also outperforms other state-of-the-art (SOTA) ATL methods i.e the cosine similarity-based approach [14], and projecting conflicting gradients (PCGrad) [13] for MTL (Table 3). For fair comparison, we also modified PCGrad by only keeping the gradients directing the common normal for auxiliary task while keeping main task gradients unchanged. We referred this method as PCGrad ATL; which also cannot outperform ours. These results prove PRF method's effectiveness and superiority as an ATG method.

Table 3 demonstrates the effectiveness of our method for ATG and the comparison shows that our method outperforms the other SOTA ATL methods. Since the proposed method is applied to the complex problem of VO, comparison is also shown with some classic VO methods (Table 4) as well as SOTA VO methods (Table 5). The superiority of geometric and unsupervised learning-based approaches in the field of VO is undeniable. We acknowledge that our results are good but clearly do not beat the SOTA VO methods. However, the goal of this paper is not to outperform all the SOTA methods of VO, rather to show that using the proposed ATG method a complex problem like monocular VO can be solved with a remarkably smaller network while maintaining a relatively competitive results (Table 5) in supervised regime. Our inference network for pose estimation has 9,438,630 number of parameters which takes about 0.031 sec in average for each prediction. It has beaten most of the existing supervised methods in memory requiring at least 5-20 times less parameters. While most of the successful supervised methods highly exploits the temporal relation between frames by utilizing long sequences i.e 3-11 along with LSTM layers feeding high resolution images i.e 1280x384 (Table 5), we use dense layers for pose estimation and only one pair of images (320x128), which is the key reason of our faster inference. To our best knowledge, no supervised learning method can achieve this level of accuracy with such small parameter space.

It is evident that our method falls behind to some extent in case of the translation error. This is because like other supervised methods, it tries to learn absolute scale automatically from training images but at such a low parameter space absolute scale is not being learn very well. Some geometric methods takes advantage of loop closure and 7-DOF alignment with ground truth for scale correction. Future research can be done utilizing additional auxiliary task like depth estimation for better learning of absolute scale.

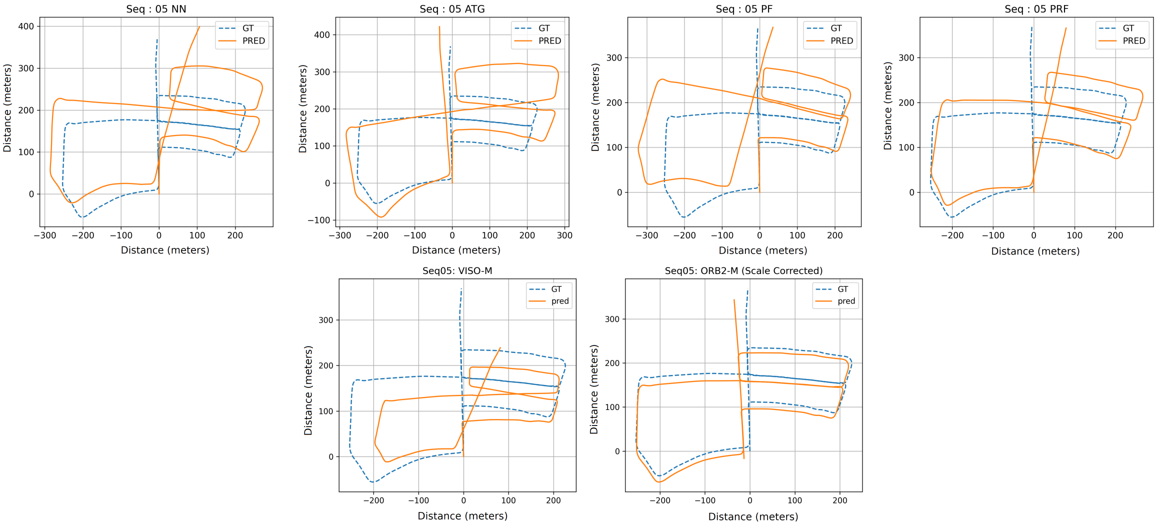


Figure 5. Comparative results of different approaches for sequence 05. Horizontal and vertical axes represent corresponding distances in the map.

Table 3. Comparison with existing methods of ATL methods

Seq.	Cosine Similarity [14]		PCGrad MTL [13]		PCGrad ATL		PF [Ours]		PRF [Ours]	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
04	16.49	1.64	17.46	4.70	18.54	0.84	11.76	0.91	15.32	1.13
05	10.86	3.36	14.04	5.21	9.99	2.74	9.47	1.89	6.90	1.50
07	6.35	3.41	16.19	10.81	7.95	2.54	8.02	3.72	6.91	2.52
10	16.82	3.13	23.35	6.94	16.90	3.24	13.54	4.21	11.60	3.47

Table 4. Comparison of our results with some classic(p) methods in the field of VO

Seq.	LIBVISO2-M (G)		ORB-SLAM(G)		DeepVO (S)[17]		UnDeep VO(U)[22]		SfmLearner (U)[23]		PRF (S) [Ours]	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
04	4.69	4.49	1.41	0.14	7.19	6.97	5.49	2.13	4.49	5.24	15.32	1.13
05	19.22	17.58	13.21	0.22	2.62	3.61	3.40	1.50	18.67	4.10	6.90	1.50
07	23.61	19.11	10.96	0.37	3.91	4.60	3.15	2.48	21.33	6.65	6.91	2.52
10	41.56	32.99	3.71	0.30	8.11	8.83	10.63	4.65	4.49	14.33	11.60	3.47

G: Geometric, S: Supervised, U: Unsupervised

Table 5. Comparison with SOTA supervised visual odometry methods

Arch.	DeepVO [17]		ESP-VO [19]		GFS-VO-RNN[20]		GFS-VO [20]		Beyond tracking[21]		PRF [Ours]	
Result	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Seq.04	7.19	6.97	6.33	6.08	5.95	2.36	**2.91**	1.30	2.96	1.76	15.32	**1.13**
Seq.05	2.62	3.61	3.35	4.93	5.85	2.55	3.27	1.62	**2.59**	**1.25**	6.90	1.50
Seq.07	3.91	4.60	3.52	5.02	5.88	2.64	3.37	2.25	**3.07**	**1.76**	6.91	2.52
Seq.10	8.11	8.83	9.77	10.2	7.44	3.19	6.32	2.33	**3.94**	**1.72**	11.60	3.47
Param.	463 M		463 M		**80 M		**47 M		**47 M		**9 M**	
Res.	1280x384		1280x384		1280x384		1280x384		1280x384		**320x128**	
Sq.len	Arbitary		Arbitary		7		7		11		**1**	

Param. : ** minimum possible parameters estimated based on available information, actual architecture may have higher number of parameters.

4. CONCLUSION

The PF method solved the issue of instability in learning. But it is prone to diminished probability which was solved by the PRF method. The PRF method additionally nullified common sources of variances successfully. Future works can include applying the proposed method for other fields and increasing the computational efficiency of the proposed methods.

REFERENCES

[1] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint*, Jun. 2017, doi: 10.48550/arXiv.1706.05098.

[2] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3070203.

[3] S. Poddar, R. Kottath, and V. Karar, "Evolution of visual odometry techniques," *arXiv preprint*, 2018, doi: 10.48550/arXiv.1804.11142.

[4] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," in *European Conference on Computer Vision*, vol. 8690, 2014, pp. 834–849, doi: 10.1007/978-3-319-10605-2˙54.

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015, doi: 10.1109/TRO.2015.2463671.

[6] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium(IVS), Proceedings*, 2011, pp. 963–968, doi: 10.1109/IVS.2011.5940405.

[7] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2011, pp. 6213–6220, doi: 10.1109/ICRA.2011.5979706.

[8] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1717–1730, 2014, doi: 10.1016/j.robot.2014.08.001.

[9] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty, "DeepVO: a deep learning approach for monocular visual odometry," *arXiv preprint*, 2016, doi: 10.48550/arXiv.1611.06069.

[10] G. Costante and T. A. Ciarfuglia, "LS-VO: learning dense optical subspace for robust visual odometry estimation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1735–1742, 2018, doi: 10.1109/LRA.2018.2803211.

[11] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: deep depth, deep pose and deep uncertainty for monocular visual odometry," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 1278–1289, doi: 10.1109/CVPR42600.2020.00136.

[12] Z. Chen, V. Badrinarayanan, C. Y. Lee, and A. Rabinovich, "GradNorm: gradient normalization for adaptive loss balancing in deep multitask networks," in *35th International Conference on Machine Learning (ICML)*, 2018, vol. 2, pp. 794–803.

[13] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020, vol. 2020-Decem.

[14] Y. Du, W. M. Czarnecki, S. M. Jayakumar, M. Farajtabar, R. Pascanu, and B. Lakshminarayanan, "Adapting auxiliary losses using gradient similarity," *arXiv preprint*, Dec. 2018, doi: 10.48550/arXiv.1812.02224.

[15] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for frame-to-frame ego-motion estimation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, Jan. 2016, doi: 10.1109/LRA.2015.2505717.

[16] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, 2015, vol. 1, pp. 486–490, doi: 10.5220/0005299304860490.

[17] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050, doi: 10.1109/ICRA.2017.7989236.

[18] N. Yang, R. Wang, J. Stückler, and D. Cremers, "Deep virtual stereo odometry: leveraging deep depth prediction for monocular direct sparse odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 817–833.

[19] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks," *International Journal of Robotics Research*, vol. 37, no. 4–5, pp. 513–542, 2018, doi: 10.1177/0278364917734298.

[20] F. Xue, Q. Wang, X. Wang, W. Dong, J. Wang, and H. Zha, "Guided feature selection for deep visual odometry," in *Asian Conference on Computer Vision (ACCV)*, vol. 11366 LNCS, 2019, pp. 293–308.

[21] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: selecting memory and refining poses for deep visual odometry," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June, pp. 8567–8575, doi: 10.1109/CVPR.2019.00877.

[22] R. Li, S. Wang, Z. Long, and D. Gu, “UnDeepVO: monocular visual odometry through unsupervised deep learning,” in *Proceedings - IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7286–7291, doi: 10.1109/ICRA.2018.8461251.

[23] L. Zhang, G. Li, and T. H. Li, “Temporal-aware SfM-learner: unsupervised learning monocular depth and motion from stereo video clips,” in *Proceedings - 3rd International Conference on Multimedia Information Processing and Retrieval, MIPR 2020*, 2020, pp. 253–258, doi: 10.1109/MIPR49039.2020.00059.

[24] P. Fischer et al., “FlowNet: learning optical flow with convolutional networks,” in Proceedings of the *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.

[26] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017, doi: 10.1109/TRO.2017.2705103.

BIOGRAPHIES OF AUTHORS

Irfan Mohammad Al Hasib is currently working as an Artificial Intelligence Engineer in Japan. He completed his B.Sc. degree from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in Mechanical Engineering in 2017. His research area is centered to the field of computer vision, deep learning, embedded system and robotics. He is always passionate about designing intelligent system in the domain of computer vision and robotics. Further info on his homepage: https://irfanhasib0.github.io/. He can be contacted at email: irfanhasib.me@gmail.com.

Sumaiya Saima Sultana is currently working as a Machine Learning Researcher in Japan. She obtained her B.Sc in Mechanical Engineering from Bangladesh University of Engineering and Technology (BUET, Bangladesh) in 2018. Her research is focused on ML quality assurance, ML robustness analysis, deep learning,computer vision and robotics. Further info on her homepage: http://sumaiyasaima05.github.io/. She can be contacted at email: sumaiyasaima.sultana@gmail.com.

Imrad Zulkar Nyeen is currently working as an Artificial Intelligence Research Engineer in Japan. He obtained his B.Sc. degree in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET, Bangladesh) in 2018. His research interests include image processing, deep learning, machine learning applications in biomedical engineering and quality assurance of AI systems. He can be contacted at email: zulkareee13@gmail.com.

Muhammad Abdus Sabur is currently working as an Artificial Intelligence Engineer in Japan who completed his B.Sc. degree from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh in Electrical and Electronic Engineering in 2018. His research area is computer vision applications, deep learning, model deployment on edge device, combinatorial optimization and signal processing. He can be contacted at email: zihad146@gmail.com.