Abstract—This paper addresses the problem of end-to-end self-supervised forecasting of depth and ego motion. Given a sequence of raw images, the aim is to forecast both the geometry and ego-motion using a self-supervised photometric loss. The architecture is designed using both convolution and transformer modules. This leverages the benefits of both modules: Inductive bias of CNN, and the multi-head attention of transformers, thus enabling a rich spatio-temporal representation that enables accurate depth forecasting. Prior work attempts to solve this problem using multi-modal input/output with supervised ground-truth data which is not practical since a large annotated dataset is required. Alternatively to prior methods, this paper forecasts depth and ego motion using only self-supervised raw images as input. The approach performs significantly well on the KITTI dataset benchmark with several performance criteria being even comparable to prior non-forecasting self-supervised monocular depth inference methods.

I. INTRODUCTION

Forecasting the future is crucial for intelligent decision making. It is a remarkable ability of human beings to effortlessly forecast what will happen next, based on the current context and prior knowledge of the scene. Forecasting sequences in real-world settings, particularly from raw sensor measurements, is a complex problem due to the exponential time-space dimensionality, the probabilistic nature of the future and the complex dynamics of the scene. Whilst much effort from the research community has been devoted to video forecasting [18], [44], [50], [64] and semantic forecasting [3], [24], [53], [57], depth and ego-motion forecasting have not received the same interest despite their importance. The geometry of the scene is essential for applications such as planning the trajectory of an agent. Anticipating is therefore important for autonomous driving auto-pilots or human/robot interaction as it is critical for the agent to quickly respond to changes in the external environment.

The first work that studied depth forecasting was carried out by Mahjourian et al. [42], the aim of that paper was to use the forecasted depth to render the next RGB image frame. They supervised the depth loss using ground-truth LiDAR scans and the warping was done using ground-truth poses. [47] used additional modalities for input, namely, a multi-modal RGB, depth, semantic and optical flow and forecasted the same future modalities. The supervision was carried out using the aforementioned ground-truth labels. [29] developed a probabilistic approach for forecasting using only input images and generated a diverse and plausible multi-modal future including depth, semantics and optical flow. However, it was supervised through ground-truth labels and the final loss was a weighted sum of future segmentation, depth and optical flow losses similar to [47]. While these methods enable forecasting the depth, they suffer from two shortcomings: [29], [42], [47] require the ground-truth labels for supervision during training.
Fig. 2. Illustration of the proposed architecture. Two sub-networks are used for training: The PoseNetwork as in network [21], [32] is used to forecast the ego-motion. The depth network combines both CNN and transformers. The Resnet34 [25] encoder extracts the spatial features for each context frame. The embedding projection module projects these features into $R^{k \times d_{model}}$ where $k = 4$ is the context frames. $N = 3$ transformer encoders are used to fuse the spatial temporal to obtain a rich spatio-temporal features. The output of the transformer module encodes the motion of the scene. The decoder uses simple transposed convolution. In order to recover the context, skip connections are pooled from the encoder. Only the last frame features are pooled for the context. The decoder outputs a disparity map that will be used along with the pose network to warp the source images onto the target.

and testing and [47] uses a multi-modal input for inference that requires either ground-truth labels or a separate network.

The work presented in this paper addresses the problem of depth and ego-motion forecasting using only monocular images with self-supervision. Monocular depth and ego-motion inference has been successful for self-supervised training [1], [9], [10], [21], [22], [31], [51], [52], [60], [66]. The basic idea is to jointly learn depth and ego-motion supervised by a photometric reconstruction loss. In this paper, it is demonstrated that it is possible to extend this self-supervised training to sequence forecasting. An accurate forecasting requires a knowledge of the ego-motion, semantics, and the motion of dynamic objects. Powered by the advances of transformers [7], [14], [15], [40], [58], and using only sensor input, the network learns a rich spatio-temporal representation that encodes the semantics, the ego-motion and the dynamic objects. Therefore avoiding the need for extra labels for training and testing. The results on the KITTI benchmark [20] show that the proposed method need for extra labels for training and testing and [47] uses a multi-modal input for inference that requires either ground-truth labels or a separate network.

The work presented in this paper addresses the problem of depth and ego-motion forecasting using only monocular images with self-supervision. Monocular depth and ego-motion inference has been successful for self-supervised training [1], [9], [10], [21], [22], [31], [51], [52], [60], [66]. The basic idea is to jointly learn depth and ego-motion supervised by a photometric reconstruction loss. In this paper, it is demonstrated that it is possible to extend this self-supervised training to sequence forecasting. An accurate forecasting requires a knowledge of the ego-motion, semantics, and the motion of dynamic objects. Powered by the advances of transformers [7], [14], [15], [40], [58], and using only sensor input, the network learns a rich spatio-temporal representation that encodes the semantics, the ego-motion and the dynamic objects. Therefore avoiding the need for extra labels for training and testing. The results on the KITTI benchmark [20] show that the proposed method need for extra labels for training and testing and [47] uses a multi-modal input for inference that requires either ground-truth labels or a separate network.

II. RELATED WORK

Previous works [21], [62], [69] have used the term “prediction” for methods that infer depth from a single image irrespective of past and future events. This could easily lead to confusion since the term “prediction” could equally be employed to refer to the prediction of future events. Therefore, the term “monocular depth inference” will be used to replace this ambiguous term. As the goal of this paper is related to time-series future prediction, the term “forecasting” will be used to refer to methods [29], [42], [47] that use observations of the past and present to predict a future depth of the scene.

A. Ego-motion and monocular depth self-supervised learning

Monocular depth inference is an ill-posed problem as an infinite number of 3D scenes can be projected onto the same 2D scene. The earliest deep learning methods [16], [17] utilized LiDAR data as ground truth for supervision. For inferring pose, [32] proposed to use a simple regression network, based on a pretrained classifier [56], supervised by an Euclidean loss. The work on spatial transformers [30] offered an end-to-end differentiable warping function, thus, enabling several works [4], [10], [21], [23], [34], [51], [52], [60], [62], [68], [69] to formulate the problem as self-supervised training of depth and pose. To account for the ill-posed nature of this task, several works have addressed different challenges. [4], [10], [52], [60] proposed to enforce depth scale and structure consistency. [23], [54] proposed more robust image reconstruction losses to handle occlusion due to moving objects. [10], [34], [51], [66] proposed multi-modal training to better supervise the network. In this paper, it will be shown that it is not only possible to infer the current depth but also forecast future depths accurately.

B. Sequence forecasting

Anticipation of the future state of a sequence is a fundamental part of the intelligent decision making process. The forecasted sequence could be a RGB video sequence [2], [18], [35], [44], [50], [64], depth image sequence [42], [63], semantic segmentation sequence [3], [12], [24], [29], [41], [53], [57] or even a multi-modal sequence [29], [47]. Early deep learning models for RGB video future forecasting leveraged several techniques including: Recurrent models [64], variational autoencoder VAE [2], generative adversarial network [44], autoregressive model [50] and normalizing flows [35]. These techniques have inspired subsequent sequence forecasting methods. Despite the importance of geometry for developing better decision making, depth forecasting is still in early development. [42] used supervised forecasted depth along with supervised future pose to warp the current image and generate the future image. Instead of using images as
input, [63] used LiDAR scans and forecast a sparse depth up to 3.0s in the future on the KITTI benchmark [20], [47] used a multi-modal input/output and forecast the depth among other modalities. [29] handled the diverse future generation by utilizing a variational model to forecast a multi-modality output. The use of multi-modalities requires additional labels or pretrained networks. This makes the training more complicated. Instead, the work presented in this paper leverages only raw images and forecasts in a self-supervised manner.

C. Vision transformers

The introduction of the Transformers in 2017 [58] revolutionized natural language processing resulting in remarkable results [7], [14], [48]. The year 2020 [8], [15] marked one of the earliest pure vision transformer networks. As opposed to recurrent networks that process sequence elements recursively and can only attend to short-term context, transformers can attend to complete sequences thereby learning long and short relationships. The multi-head attention could be considered as fully connected graph of the sequence’s features. It demonstrated its success by outperforming convolution based networks on several benchmarks including classification [15], [37], [67], detection [8], [36], [37] and segmentation [11], [40]. The has led to a paradigm shift [39], transformers are slowly [37], [67], [69], detection [8], [36], [37] and segmentation [11], [40].

The forecasting network is composed of two sub-networks: a pose network forecasts 6 parameters using the axis-angle error of \( \beta \). The pose network forecasts \( \beta \) of \( \beta \) is assumed to follow a Gaussian distribution, the maximization problem:

\[
\pi = \arg \max \sum_{m} P_{model}(I_{t+n}^{m}) \tag{5}
\]

m is the number of samples. If \( P_{model} \) is assumed to follow a Laplacian distribution \( P_{model} \sim \text{Lap}(\mu = I_{t+n}^{m}; \beta = \sigma^{2}) \). \( I_{t+n}^{m} \) is the warped image. Maximizing the Eq. 5 is equivalent to minimizing an L1 error of \( I_{t+n}^{m} \) and the known image frame \( I_{t+n} \). Similarly, if the distribution is assumed to follow a Gaussian distribution, the maximization is equivalent to minimizing an L2 error.

B. The architecture

The architecture of the network is depicted in Fig. 2. The forecasting network is composed of two sub-networks: a pose net to forecast the future \( I_{t+n} \) and a depth network that forecast \( D_{t+n} \) (see Eq. 1). Similar to [32], the pose-net is composed of a classification network [25] as a feature extractor followed by a simple pose decoder as in [21]. The pose network forecasts 6 parameters using the axis-angle representation. The depth network leverages a hybrid CNN and Transformer network as in [8] that is designed to forecast the geometry of the scene. This network benefits from both modules. The convolution module is used to extract the spatial features of the frames as it is memory efficient, easy to train and does not require large pretraining. The transformer module is used for better temporal feature aggregation. The multi-head attention could be considered as a fully-connected graph of the features of each frame. Therefore the information is correlated across all the frames rather than incrementally, one step at a time, as in LSTM [27]. The architecture consists of three modules: an encoder, temporal aggregation module and a decoder.

1) Encoder: ResNet [25] is one of the most used foundation models [5]. It has demonstrated its success as a task agnostic feature extractor for nearly all vision tasks. In this work, ResNet34 is used as feature extractor. It is pretrained on ImageNet [13] for better convergence. Each context frame is fed-forward and a pyramid of features is extracted. These features encode the spatial relationship between each scene separately. Thus, at the output of this module, a pyramid of

\[
(f_{x}, f_{y}, c_{x}, c_{y}) \quad \text{are the camera intrinsic parameters.} \quad \text{H is the homogeneous coordinates transformation and} \quad ^{s}\mathbf{T}_{tar} \in \mathbb{R}^{3\times4} \quad \text{is the ego-motion. Similarly, a pixel} \quad \mathbf{p} \quad \text{is projected to a 3D point} \quad \mathbf{D}(\mathbf{p}) \quad \text{using the spatial transformers} \quad [30]. \quad \text{The reverse warping defined here is fully differentiable and uses a bilinear interpolation. Similarly, the self-supervised depth and ego-motion forecasting is defined here by considering the source frames} \quad \mathbf{I}_{s} \quad \text{and the target} \quad \mathbf{I}_{t+n} \quad \text{with the corresponding poses} \quad ^{t}\mathbf{T}_{t+n} : \forall i \in \{t-k, \ldots, t\} \quad \text{The frame} \quad \mathbf{I}_{t+n} \quad \text{is obtained by reverse warping the context frames.}

\text{Reconstructing the frame} \quad \mathbf{I}_{t+n} \quad \text{using the depth and the pose from only the context by a warping could be formulated as a maximum likelihood problem:} \quad \text{Eq. 5 is equivalent to minimizing an L1 error of} \quad \mathbf{I}_{t+n} \quad \text{and the known image frame} \quad \mathbf{I}_{t+n}. \quad \text{Similarly, if the distribution is assumed to follow a Gaussian distribution, the maximization is equivalent to minimizing an L2 error.}
spatial features for each frame is constructed. These features will be correlated temporally using the Temporal aggregation module TAM.

2) Temporal aggregation module: Since its introduction, transformer have demonstrated their performance outperforming their LSTM/RNN counterparts in various sequence learning benchmarks [7], [14], [49], [59], forecasting accurate depth requires knowledge of the static objects, accurate ego-motion and knowledge of the motion of the dynamic objects. The last layer of the encoder is assumed to encode higher abstraction features (e.g. recognizing objects). Therefore, correlating temporally these features allows the extraction of the motion features of the scene. The TAM consists of two sub-modules:

- **Embedding projection:** The dimensions after flattening the feature output of the last layer of the encoder is not memory efficient for the transformers. The embedding projection maps these features as: \( R^{K \times C \times H \times W} \rightarrow R^{K \times d_{asc}} \).

- **Transformer encoder:** After projecting the features using the embedding layer, a Transformer encoder with \( N \) layer, \( m \) multi-head attention and \( d_{enc} \) is used. It correlates the spatial features of the sequence producing a spatio-temporal fused features.

3) **Depth decoder:** After the spatio-temporal fusion, the decoder takes these spatio-temporal features along with the context features as input and decodes them to produce a disparity map. As depicted in Fig. 2, the context of the scene is obtained by pooling the features of last frame in the encoder. Two levels are pooled and concatenated. \( dec_{l+n} = [enc_{f_{l}}, TAM(enc_{f_{l-t+l}})] \). The high level features (skip connection before the TAM) enable learning the motion while the low level features (skip connection at the start of ResNet) recover the finer details lost by the down-sampling. Therefore, the decoder maps (context + motion \( \rightarrow \) depth).

Each level of the decoder consists of a simple sequential layer of: transposed convolution with kernel of \( 3 \times 3 \) with similar channels as the encoder, batch normalization and Relu activation in that order. The forecasting head consists of a convolution with a kernel of \( 1 \times 1 \) and Sigmoid activation. The output of this activation, \( \sigma \), is re-scaled to obtain the depth \( D = \frac{1}{\sigma + \epsilon} \), where \( a \) and \( b \) are chosen to constrain \( D \) between 0.1 and 100 units, similar to [21]. For training, each level has a forecasting head but only the last head is used for inference.

C. **Objective functions**

As formulated in Sec. III-A, learning the parameters \( \Theta \) involves maximizing the maximum likelihood of \( P_{model} \). Similar to the prior work, along with the \( L_1 \) error other additional regularization terms are used and defined as:

- **Photometric loss:** Following [21], [52], [69] The photometric loss seeks to reconstruct the target image by warping the source images using the forecast pose and depth. An \( L_1 \) loss is defined as follows:

  \[
  L_{rec}(\hat{I}_{l+n}, I_{l+n}) = \sum_p |I_{l+n}(p) - \hat{I}_{l+n}(p)|
  \]  

(6)

where \( \hat{I}_{l+n}(p) \) is the reverse warped target image obtained by Eq. 2. This simple \( L_1 \) is regularized using SSIM [61] that has a similar objective to reconstruct the image. The final photometric loss is defined as:

\[
L_{pe}(I_{l+n}, \hat{I}_{l+n}) = \sum_p \left[ (1 - \alpha) \text{SSIM}(I_{l+n}(p), \hat{I}_{l+n}(p)) + \alpha |I_{l+n}(p) - \hat{I}_{l+n}(p)| \right]
\]  

(7)

- **Depth smoothness:** An edge-aware gradient smoothness constraint is used to regularize the photometric loss. The disparity map is constrained to be locally smooth through the use an image-edge weighted \( L_1 \) penalty, as discontinuities often occur at image gradients. This regularization is defined as [26]:

\[
L_s(D_{l+n}) = \sum_p \left[ |\partial_x D_{l+n}(p)| e^{-|\partial_x I_{l+n}(p)|} + |\partial_y D_{l+n}(p)| e^{-|\partial_y I_{l+n}(p)|} \right]
\]  

(8)

Training with these loss functions is subject to major challenges: gradient locality, occlusion and out of view-objects. Gradient locality is a result of bilinear interpolation [30], [69]. The supervision is derived from the four neighbors of \( I(p_t) \) which could degrade training if that region is low-textured. Following [19], [21], [22], an explicit multi-scale approach is used to allow the gradient to be derived from larger spatial regions. A forecasting head is used at each level to obtain each level’s disparity map during training. Eq. 2 assumes global ego-motion to calculate the disparity. Supervising directly using this objective is inaccurate when this assumption is violated (e.g. the camera is static or a dynamic object moves with same velocity as the camera). According to [21] this problem can manifest itself as ‘holes’ of infinite depth. This could be mitigated by masking the pixels that do not change the appearance from one frame to the next. A commonly used solution [21], [69] is to learn a mask \( \mu \) that weigh the contribution of each pixel, while [69] uses an additional branch to learn this mask. This paper uses the auto-masking defined in [21] to learn a binary mask \( \mu \) as follows:

\[
\mu(I_{l+n}, \hat{I}_{l+n}, I_t) = L_{pe}(I_{l+n}, \hat{I}_{l+n}) < L_{pe}(I_{l+n}, I_t)
\]

(9)

\( \mu \) is set to only include the loss when the photometric loss of the warped image \( \hat{I}_{l+n} \) is lower than the original unwarped image \( I_t \). The final objective function is defined as:

\[
\mathcal{L} = \sum_l [ \mu L_p + \alpha_d L_s ]
\]

(10)

where \( l \) is the scale level of the forecasted depth.

IV. **Experiments**

A. **Setting**

1) **KITTI benchmark** [20]: Following the prior work [17], [21], [38], [43], [60], [65], [69], the Eigen et al. [17] split is used with Zhou et al. [69] pre-processing to remove static frames. Frames without sufficient context are excluded from the training and testing. This split has become the de facto benchmark for training and evaluating depth that is used by nearly all depth methods.
Table I shows the results of the proposed method on the KITTI benchmark [20]. As specified in Sec. IV-A2, the method is compared to three approaches: prior work on depth inference; copying last frame; and adapting monodepth2 [21] for future forecasting. The proposed method outperforms the forecasting baselines for both short and mid-term forecasting especially for short range forecasting. The results are even comparable to non-forecasting methods [17], [38], [65], [69] that have access to I_{t+1:yn}. The gap between state-of-the-art depth inference and the proposed forecasting method is reasonable due to the uncertainty of the future, the unobservability of certain events such as a new object entering the scene and the complexity of natural videos that requires modeling correlations across space-time with much higher input dimensions.

Fig. 1 shows an example of depth forecasting on Eigen test split. Several observation can be made:

- The network handles correctly the out-of-view object (i.e. the bicycle on the scene).
- The network learned the correct ego-motion: The position of the static objects is accurate.

These results suggest that the network is able to learn a rich spatio-temporal representation that enable learning the motion, geometry and the semantics of the scene. Thus, extending the self-supervision depth inference to perform future forecasting with comparable results. A further analysis is done to evaluate and validate the choices of the network in Sec. IV-D.

2) Baselines: As discussed above, previous work on depth forecasting has been supervised using LiDAR scans, and has used a multimodal network that provides depth. Their evaluation is not performed on the Eigen split, nor does it use the defacto self-supervised metrics. In order to fairly evaluate the proposed method, a self-supervised monocular formulation will be used to compare performance with the KITTI Eigen split benchmark. Comparisons will be made with three approaches: prior work on self-supervised depth inference [17], [21], [38], [60], [65], [69]; copy of the last observed LiDAR frame as done in [47]; and ForecastMonodepth2, a modified version of [21] that is adapted for forecasting pose/depth.

3) Evaluation metrics: For evaluation, the metrics of previous works [17] are used for the depth. During the evaluation, the depth is capped to 80m. To resolve the scale ambiguity, the forecasted depth map is multiplied by median scaling where $s = \frac{\text{median}(D)}{\text{median}(D_{gt})}$. For the pose evaluation, the Absolute Trajectory Error (ATE) defined in [55] is evaluated on KITTI odometry benchmark [20] sequences 09 and 10.

4) Implementation details: PyTorch [46] is used for all models. The networks are trained for 20 epochs, with a batch size of 8. The Adam optimizer [33] is used with a learning rate of $lr = 10^{-4}$ and $(\beta_1, \beta_2) = (0.9, 0.999)$. As training proceeds, the learning rate is decayed at epoch 15 to $10^{-5}$. The SSIM weight is set to $\alpha = 0.15$ and the smoothing regularization weight to $\alpha_d = 0.001$. $l = 4$ scales are used for each output of the decoder. At each scale, the depth is upsampled to the target image size. $d_{model} = 2048$, $m = 16$ and $N = 3$ for the TAM projection. The input images are resized to $192 \times 640$. Two data augmentations were performed: horizontal flips with probability $p = 0.5$ and color jitter with $p = 1$. $k = 4$ frames are used for context sequence and $n = 5$ is used for short term forecasting and $n = 10$ for mid-term forecasting as in [47] which corresponds to forecasting 0.5s and 1.0s into the future. The ForecastedMonodepth2 is the same as [21] with a modified input. The context images are concatenated and used as input for both depth and pose networks.
| Method                      | Abs Rel | Sq Rel | RMSE  | log RMSE |
|-----------------------------|---------|--------|-------|----------|
| Ours                        | 0.178   | 1.645  | 0.257 | 6.196    |
| Without TAM                 | 0.205   | 1.745  | 0.296 | 6.565    |
| Shared pose/depth features  | 0.208   | 1.745  | 0.296 | 6.565    |
| Single scale                | 0.208   | 1.950  | 0.283 | 6.595    |
| Disable auto-masking        | 0.193   | 1.774  | 0.273 | 6.374    |

TABLE III
ABLATION STUDY OF THE DIFFERENT COMPONENT OF THE PROPOSED METHOD. THE EVALUATION WAS DONE ON KITTI BENCHMARK USING EIGEN SPLIT [17]

C. Ego-Motion forecasting results

Table II shows the results of the proposed network on the KITTI odometry benchmark [20]. Similar to depth, the assessment is made by comparing with non-forecasting prior works. To avoid data leakage, the network is trained from scratch on the sequences 00-08 of the KITTI odometry benchmark. The network takes only the context images \( I_n \) and forecasts \( \{T_{t+n}\} \). The ATE results in Table II show the proposed network achieves a competitive result relative to other non-forecasting approaches. All the methods are trained in the monocular setting, and therefore scaled at test time using ground-truth. These results suggest that using the proposed architecture along with the self-supervised loss function successfully learns the future joint depth and ego-motion.

D. Ablation study

To further analyse the network, several ablations are made. Table III depicts a comparison of the proposed model with several variants. The evaluation is done for short-term forecasting \( n = 5 \) using \( k = 4 \) context frames.

1) **Effect of the Temporal Aggregation Module:** In order to evaluate the contribution of the multi-head attention, a variant of the proposed method is designed by replacing the TAM module by a simple concatenation of the last layer features. From Table III, the improvement induced by the TAM module is significant across all metrics. These results suggest that the performance obtained by the proposed method is achieved through the TAM module. Since the TAM aggregates the temporal information across all frames using a learned attention, the temporal features are better correlated and the final representation successfully encodes the spatio-temporal relationship between the images.

2) **Effect of sharing the encoder of depth and ego-motion:** Since both pose and depth networks encode the future motion and geometry of the scene, it is expected that sharing the encoders of these networks yield better results. However, as reported by Table III, the degradation is significant. Even though these tasks are collaborative, sharing the encoder will result in a set of parameters \( \hat{\theta} \) that are neither the best local optima for the depth nor for the pose. By alleviating this restriction and separating the encoders, the network learns better local optima for both pose and depth.

3) **The benefit of using multiple scales:** In order to evaluate the multi-scale extension, a variant of the proposed method that uses only one scale is trained. As illustrated in the Table III The network benefits from the multi-scale. The reverse warping uses bi-linear interpolation. As mentioned earlier, each depth point depends only on the four neighboring warped points. By using a multi-scale depth at training-time the gradient is derived from a larger spatial region directly at each scale.

4) **Effect of auto-masking:** Table III compares the proposed method with a variant without using the auto-masking defined in Sec. III-C. The results show that using auto-masking improves all four evaluation criteria. This demonstrates that using auto-masking, for pixels that do not change appearance, reject these outliers that inhibit the optimization. This leads to better accuracy of the forecasted depth.

E. Limitations and perspective

Even though the proposed forecasting method yields good results, there exists a gap with respect to non-forecasting methods. Several limitations contribute to this:

- A common assumption across SOTA methods is that the environment is deterministic and that there is only one possible future. However, this is not accurate since there are multiple plausible futures. Given the stochastic nature of the forecasting proposed here, the network will tend to forecast the mean of all the possible outcomes [2].
- The network does not forecast the correct boundaries of the objects. This is due to the formulation as a maximum likelihood problem with a Laplacian distribution assumption and the deterministic nature of the architecture. As a result, the boundaries of the dynamic objects are smoothed.
- Due to the problem formulation, the scale of the forecasted depth is ambiguous.
- The model fails to account for the motion of distant dynamic objects due to lack of parallax. and fine objects are ignored by the network.

To address these limitations future work will investigate constraining depth to be conditional on the input image structure as in [6]. Concerning scale, one solution could be to infer directly from the scene as in [4], [10], [60].

V. Conclusion

This paper proposed an approach for forecasting future depth and ego motion using only raw images as input. This problem is addressed as end-to-end self-supervised forecasting of the future depth and ego motion. Results showed significant performances on several KITTI dataset benchmarks [20]. The performance criteria are even comparable with non-forecasting self-supervised monocular depth inference methods [17], [38], [65], [69]. The proposed architecture demonstrates the effectiveness of combining the inductive bias of the CNN as a spatial feature extractor and the multi-head attention of transformers for temporal aggregation. The proposed method learns a spatio-temporal representation that captures the context and the motion of the scene.  

---

2This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-01101931). The authors would like to acknowledge the Association Nationale Recherche Technologie (ANRT) for CIFRE funding (n2019/1649).
REFERENCES

[1] Yasin Almalioglu, Muhamad Risqi U. Saputra, Pedro P.B. De Gusmao, Andrew Markham, and Niki Trigoni. GVANO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In Proceedings - IEEE International Conference on Robotics and Automation, volume 2019-May, pages 5474–5480, 2019.

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy Campbell, and Sergey Levine. Stochastic variational video prediction. In 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018.

[3] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. Bayesian prediction of future street scenes using synthetic likelihoods. In 7th International Conference on Learning Representations, ICLR 2019.

[4] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Ge, Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunh
