3D Shape representation has substantial effects on 3D shape reconstruction. Primitive-based representations approximate a 3D shape mainly by a set of simple implicit primitives, but the low geometrical complexity of the primitives limits the shape resolution. Moreover, setting a sufficient number of primitives for an arbitrary shape is challenging. To overcome these issues, we propose a constrained implicit algebraic surface as the primitive with few learnable coefficients and higher geometrical complexities and a deep neural network to produce these primitives. Our experiments demonstrate the superiorities of our method in terms of representation power compared to the state-of-the-art methods in single RGB image 3D shape reconstruction. Furthermore, we show that our method can semantically learn segments of 3D shapes in an unsupervised manner. The code is publicly available from this link.
shapes. Contrastingly, primitive-based methods approximate 3D shapes by a group of primitives such as cubes [40], ellipsoids [12], superquadrics [30], and convexes [10]. Despite their advantages in visualization and direct usage for various applications, the resolutions of reconstructed shapes are limited due to the simple topology (i.e., genus-zero) of the primitives. Consequently, approximating a 3D shape requires many of these simple primitives. Moreover, since the geometrical complexity varies from shape to shape, determining a sufficient number of primitives is challenging for an arbitrary shape.

In this paper, we propose a novel primitive-based 3D shape representation based on the learnable implicit algebraic surfaces named 3DIAS as shown in Figure 1. Since implicit algebraic surfaces have high degrees of freedom, they can describe complex shapes better than simple primitives [2]. Besides, identifying an implicit algebraic primitive is straightforward and depends on only a few parameters. We apply various constraints on these primitives to facilitate learning and achieve detailed appearances. We limit our primitives to the class of algebraically solvable implicit algebraic surfaces to assist fast 2D rendering and 3D visualization, which can be useful in many computer graphics applications. Furthermore, we develop an upper bound constraint with an efficient parameterization to guarantee that the primitives have closed surfaces and controlled sizes. Finally, we guide the primitives to cover different segments of a target shape by restricting the locations of their centers. To generate these primitives, we design a DNN-based encoder-decoder that captures the information of an observation (e.g., single image) and provides the parameters of the primitives. In our experiments, we show that our method outperforms state-of-the-art methods with most of the metrics. Moreover, we experimentally demonstrate that 3DIAS can semantically learn the components of 3D shapes without any supervision and adjust the number of used primitives.

We summarize, our main contributions as follows:

- We propose a novel primitive-based 3D shape representation with the learnable implicit algebraic surfaces, which can produce more complex topologies with few parameters hence appropriate for describing geometrically complex shapes.
- We develop various constraints to produce solvable and closed primitives with proper scales in desired locations to ease learning and generate appealing results.
- We experimentally demonstrate that 3DIAS outperforms state-of-the-art methods. Furthermore, we show that it can semantically learn the components of 3D shapes and adjust the number of used primitives.

2. Related Work

In this section, we review some related DNN-based 3D shape reconstruction methods with various representations. Explicit representations. A set of voxel is commonly used for discriminative [25, 32] and generative [9, 13] tasks since it is the most simple way in 3D representation. However, the results represented with voxel have a limitation in resolution due to memory issues. Although [17, 38] proposed to reconstruct 3D objects in a multi-scale fashion, they are still limited to comparably small $256^3$ voxel grids and require multiple forward passes to generate final 3D voxels. 3D point clouds give an alternative representation of 3D shapes. Qi et al. [31, 33] pioneered point clouds as a discriminative representation for 3D tasks using deep learning. Fan et al. [11] introduced point clouds in 3D reconstruction task as an output representation. However, since point clouds have no information for connections between the points, it needs additional post-processing procedures [3, 5] to build the 3D mesh as an output. Mesh is another commonly used representation for 3D shape [14, 20]. However, most of
methods in 3D shape reconstruction using meshes generate meshes with simple topology [41] or utilize a template as a reference. They can only manage the objects from the same class [20, 24] and can not guarantee to produce closed surfaces [14].

**Implicit Representations.** Implicit representation is a good alternative to avoid the problems above. In contrast to the mesh-based approaches, implicit-based methods do not require a template from the same object class as input. There are mainly two approaches; isosurface-based and primitive-based methods. Chen et al. [8] propose a neural network that takes the 3D points and latent code of the shape, then outputs the values for each point, indicating whether the point is outside or inside the shape as an occupancy function. Park et al. [29] utilize the signed distance function to obtain the zero-set surface of the shape. Tatarchenko et al. [39] proposed the occupancy function that implicitly describes the 3D surfaces as the continuous decision boundary of a DNN-based classifier. Compared to voxel representation, this method can estimate an occupancy function continuous in 3D with a learnable neural network that can generate at any arbitrary resolution. This approach significantly decreases memory usage during training. The surface can be extracted as a mesh representation from the learned model at test time by a multi-resolution isosurface extraction method. The finite resolution of the voxel or octree cells limits the accuracy of the reconstructed shape by these methods and their ability to capture fine details of 3D shapes. Deng et al. [10] represented a shape with a convex combination of half-planes. These methods use a single global latent vector to represent entire surfaces of a 3D shape. The latent vector is decoded into continuous surfaces with the corresponding implicit networks. While this technique successfully models geometry, it often requires many primitives to obtain a desirable appearance, and it is unclear how many primitives are required.

For the 3D reconstruction task, we compare our implicit-based approach against several state-of-the-art implicit-based methods such as Structured Implicit Function (SIF) [12], OccNet [26], and CvxNet [10]. Moreover, we select Pixel2Mesh [41] and AtlasNet [15], which use explicit surface generation in contrast to the previous methods.

3. 3DIAS

In this section, we first introduce our 3D shape representation based on implicit algebraic surfaces. Then we explain the additive constraints for effective learning. Next, we describe the proposed network and learning procedure to reconstruct the surface of 3D shapes with our representation.

3.1. 3D shape representation

3.1.1 Implicit algebraic primitives

We build a complex target 3D shape with a combination of primitives \( p(x, y, z) \) that are the building blocks of 3D (i.e., basic geometric forms) as shown in Figure 1. To select a primitive with a large degree of freedom (i.e., complex geometry and topology) and few parameters, we employ the implicit algebraic surface that is a zero-level set of a multivariate polynomial function of \( x, y, \) and \( z \) as Eq. 1:

\[
p(x, y, z) = \sum_{0 \leq i+j+k \leq d} a_{ijk} x^i y^j z^k = v^T A v = 0, \tag{1}
\]

where \( v = [1, x, y, z, x^2, y^2, z^2, \ldots] \), and \( d, a_{ijk}, \) and \( A \) are the degree, the coefficients, and coefficient matrix of the polynomial function, respectively. Like many other implicit
surfaces, the implicit algebraic surface divides the space and maps points in 3D space into negative and positive values. Therefore, to represent detailed surfaces \( S(x, y, z) \) of 3D shapes, we can combine these primitives by utilizing constructive solid geometry [19] and apply boolean operations to them, which can be formulated as Eq. 2:

\[
S(x, y, z) = \bigcup_{m=1}^{M} p_m(x, y, z) = \min(p_1(x, y, z), \ldots, p_M(x, y, z)),
\]

where \( M \) is the number of primitives in the union.

### 3.1.2 Constraints on primitives

We apply a set of constraints on the defined implicit algebraic primitive \( p(x, y, z) \) to better approximate the surface \( S(x, y, z) \) of a target 3D shape.

#### Solvability of primitives. Easy visualization and rendering attributes for a 3D shape representation can be useful in many computer graphics and virtual reality applications. The class of implicit algebraic primitives with algebraic solutions are appropriate representations for ray-tracing hence achieving these properties. Accordingly, we use multivariate quartic \( (d = 4) \) polynomial functions as the primitives \( p(x, y, z) \) because they have the highest degree of freedom among all implicit algebraic primitives with closed-form algebraic solutions [34].

#### Closedness and scales of primitives. Beyond the aforementioned constraint, we need to guarantee that the reconstructed shape and hence all primitives have closed surfaces as Figure 2. We can ensure that a quartic primitive has a closed surface by enforcing its fourth-degree terms to always be positive [22] as Eq. 3:

\[
p^4(x, y, z) = \sum_{i+j+k=4} a_{ijk} x^i y^j z^k = u A_{[5:10]}^T u^T > 0,
\]

where \( u = [x^2, y^2, z^2, xy, yz, zx] \) and \( A_{[5:10]} \) is the \( 6 \times 6 \) sub-matrix of \( A \), including the coefficients of fourth-degree terms. This implies that \( A_{[5:10]} \) is a positive definite (PD) matrix. Note that, with the PD matrix \( A_{[5:10]} \), the algebraic surface exists if and only if \( p^4(x, y, z) = \sum_{0 \leq i+j+k \leq 3} a_{ijk} x^i y^j z^k \) is negative and \( |p^4(x, y, z)| > |p^4(x, y, z)| \) for some points in \( \mathbb{R}^3 \). Otherwise, the primitive has zero volume because it has no real-valued solution.

Moreover, since each primitive reconstructs a different segment of a target 3D shape, we need to ensure that its volume is smaller than the target shape. To prevent generating large primitives and control their scales, we develop an upper bound for each primitive. To reconstruct a primitive \( p(x, y, z) \) included in the upper bound \( q(x, y, z) \), it is sufficient to satisfies the inequality \( q(x, y, z) < p(x, y, z) \) in \( \mathbb{R}^3 \) which is also equivalent to Eq. 4:

\[
p(x, y, z) = h(x, y, z) + q(x, y, z),
\]

where \( h(x, y, z) \) is a positive-valued function. The function \( h(x, y, z) \) is always positive if and only if its matrix of coefficients \( H \) be positive definite. As a result, the coefficient matrix \( A \) of primitive \( p(x, y, z) \) is the summation of a positive definite matrix \( H_{10 \times 10} \) and the coefficient matrix \( Q_{10 \times 10} \) of the upper bound \( q(x, y, z) \). For simplicity we consider the upper bound \( q(x, y, z) \) as Eq. 5:

\[
q(x, y, z) = x^4 + y^4 + z^4 - R,
\]

where \( R \) is a positive value that controls the size of the upper bound. Note that the developed upper bound is a more general constraint that also holds the criteria for the closedness constraint. For proof, refer to the supplementary materials.

#### Locations of primitives. We also encourage the primitives to better cover different components of 3D shape by applying a constraint on their locations. Accordingly, we restrict the locations of their centers \( c = (c_1, c_2, c_3) \) into the areas nearby the shape and reformulate the primitives as Eq. 6:

\[
p(x, y, z) = \sum_{0 \leq i+j+k \leq 4} a_{ijk} x^i y^j z^k = 0.
\]

Therefore, within these constraints, we can reconstruct primitives with controlled scales and locations, which facilitates the reconstruction and provides more details.

### 3.2 3D shape reconstruction

To reconstruct a 3D shape with the proposed representation of an input observation \( o \in X \) (e.g., single image), we design a DNN architecture that receives the input and outputs the corresponding matrix \( H \), center \( c \), and parameter \( R \) for each primitive as shown in Figure 3.

### 3.2.1 Training losses

We apply various losses to reconstruct 3D shapes.

#### Loss sign. Since the target surface in 3D space divides the inside and outside, we define a sign function \( \text{sign}(x, y, z) : \mathbb{R}^3 \rightarrow \{0, -1, 1\} \) on sample points \( P \subset \mathbb{R}^3 \) where the values \( 0, -1, 1 \) correspond to the points on the target surface, its inside, and its outside, respectively. Likewise, we can classify the points on/inside/outside the reconstructed implicit surfaces \( S(x, y, z) \) and reduce a loss between their predicted and ground truth signs. We use mean square error (MSE) as the loss function as Eq. 7:

\[
\mathcal{L}_{sign}^R = \sum_{i \in \{\text{on, in, out}\}} \lambda_i \cdot |\text{sign}(p_i) - \text{sign}(p_i)|^2,
\]
where $B \subset \mathcal{X}$ and $\lambda_i$ are a training batch and the weights corresponding to each sign, respectively. Note that $\mathcal{L}^{\text{sign}}_B$ enforce the network to reconstruct the desired surface and refuse to generate the redundant surfaces simultaneously. Moreover, the MSE loss forces further attention on distinguishing the inside and outside points near the reconstructed surface because their $\tanh S(p_i)$ are near zero while their ground truths are $-1$ and $+1$, respectively.

**Loss normal.** To improve the reconstruction, we use the normal vectors as second-order information. Therefore, we define a MSE loss between the ground-truth normal vectors of the sample points $p_{on}$ on the surface of a target mesh model $n_b$ and their normal vectors $n_i$ obtained by Eq. 8:

$$
\mathcal{L}^n_B = E_{p_{on} \sim \mathcal{P}} \| n_i(p_{on}) - n_b(p_{on}) \|^2, \quad (8)
$$

where the normal vectors on the union surface can be directly determined for any point on the surface as Eq. 9:

$$
\begin{align*}
    n_i &= \frac{\nabla S(x, y, z)}{||\nabla S(x, y, z)||} \equiv \frac{\left( \frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial z} \right)}{\sqrt{\left( \frac{\partial S}{\partial x} \right)^2 + \left( \frac{\partial S}{\partial y} \right)^2 + \left( \frac{\partial S}{\partial z} \right)^2}}, \\
    \text{subject to:} \quad &\nabla S(x, y, z) = \nabla p_{m^*}(x, y, z), \\
    &m^* = \arg \min_m (p_m(x, y, z)).
\end{align*} 
\quad (9)
$$

Note that $m^*$ is the index of the closest primitive (i.e., the primitive with the smallest value $p_{m^*}(x, y, z)$) to the point.

Finally, the total loss $\mathcal{L}^{\text{total}}_B$ is the weighting average of all defined losses with the corresponding weights as Eq. 10:

$$
\mathcal{L}^{\text{total}}_B = \mathcal{L}^{\text{sign}}_B + \lambda_n \mathcal{L}^n_B. \quad (10)
$$

### 3.2.2 Implementation details

We consider the bounding box $C = [-1, 1]^3$ and fit the given input 3D shapes into it by keeping its aspect ratio. Then we extract 1M points from $[-1.1, 1.1]^3 \subset \mathbb{R}^3$ surrounding the 3D shapes, and at each iteration, we randomly select 1% of them as jointly inside points $p_{in}$ and outside points $p_{out}$ for all shapes in the batch $B$. Moreover, we pick 10k points $p_{on}$ on the surface of each 3D shape and randomly select 20% of them at each iteration.

To capture the information of the input observation $o \in \mathcal{X}$ we employ the pretrained ResNet-18 [18] as the encoder. Then three sets of independent fully connected (FC) layers $(4096, 4096, 4096, 55 \times M), (1024, 512, 256, 3 \times M), (256, M)$ decodes the encoded features to obtain the parameters of each symmetric matrix $B$, scalar $R$ and center $c$ for $M = 100$ primitives in Eq. 2, respectively. All FC layers except the last layers are empowered by the ReLU non-linear activation function. We also apply three batch normalization layers after the first three FC layers of $B$ decoder to accelerate training and boost the performance.

To ensure $H$ is a PD matrix, we parameterize it as Eq. 11:

$$
H = BB^T + \alpha I \succ 0 \quad (11)
$$

where $\alpha = 0.0001$ is a small scalar factor, and $B$ and $I$ are $10 \times 10$ symmetric and identity matrices, respectively. Moreover, we apply the $\text{sigmoid}$ function on the output of the $R$ decoder to generate a value in $(0, 1) \subset \mathbb{R}^3$ as the parameter $R$ of Eq. 5 to control the size of all primitives and guarantee that they are not larger than the size of the bounding box $C$. Furthermore, we apply $\tanh$ on the output of the $c$ decoder to generate the centers inside the bounding box $C$. Therefore, each primitive is parameterized with 59 parameters in total including three parameter for the center $c = (c_1, c_2, c_3)$, one parameter for the $R$, and 55 parameters for the matrix $B$.

We set the parameters $\lambda_{on}, \lambda_{in}, \lambda_{out}$, and $\lambda_n$ as 2, 1, 10, and 1, respectively. We train our encoder-decoder architecture with Adam optimizer with the initial learning rate $1e^{-4}$, weight decay $1e^{-7}$, and batch size of 64. We implement our model in Python3.7 using PyTorch via CUDA instruction.

### 4. Experiments

In this section, we provide information about the evaluation setups and show qualitative and quantitative results of our method compared to state-of-the-art methods on single RGB image 3D shape reconstruction. We also perform various ablation studies to analyze our method better. More experiments are available in the supplementary material.

#### 4.1. Dataset and Metrics

We evaluate our approach on the subset of the ShapeNet dataset [6] with the same image renderings and training/testing split provided by Choy et al. [9]. We also employ mesh-fusion [36] to generate watertight meshes from the 3D CAD models. We then use Houdini [35] to extract the inside/outside/on points and the normal vectors. For evaluation, we use the volumetric IoU, Chamfer [11], and F-Score [23] metrics. Volumetric IoU is used to measure the overlapped volume between the ground truth meshes and the reconstructed surfaces. Chamfer is the mean of the accuracy and the completeness score. The mean distance of points on the reconstructed surface to their nearest neighbors on the ground truth mesh is defined as the accuracy metric. The completeness metric is defined in the opposite direction of the accuracy metric. F-score is the harmonic mean of precision which shows the percentage of correctly reconstructed surfaces. To compute IoU, we sample 100k points from the bounding box. To evaluate Chamfer and F-score, we first transfer the reconstructed surfaces to meshes, then similar to CvxNet[10] we sample 100k points on the reconstructed and the ground-truth meshes.
We experimentally evaluate our method 3DIAS trained on multi-class and compare it with state-of-the-art methods on single RGB image 3D shape reconstruction and summarize the results in Table 1. The experiments demonstrate the superiority of 3DIAS compared to the explicit-based methods P2M [41], AtlasNet [15], OccNet [26], SIF [12], and CvxNet [10] in terms of volumetric IoU and F-score. We also achieve the second-best performance with the Chamfer metric. We show more quantitative results of 3DIAS for the trained network on single-class in the supplementary material.

Moreover, we qualitatively evaluate 3DIAS trained on single-class and compare it with the previous methods in Figure 4. The results illustrate that 3DIAS achieves smooth surfaces with desirable geometrical details. 3DIAS, unlike the previous methods, is successful in reconstructing the 3D shape with more complex topologies (e.g., chair) as shown in Figure 4a. Moreover, compared to CvxNet [10], 3DIAS can better reconstruct thin shapes (e.g., lamp) and when similar shapes are rare in the training dataset (e.g., airplane and car), see Figure 4b.

### 4.2. Ablation study

We perform several ablation studies to analyze our proposed representation and reconstruction procedure. First, we show the ability of our method to generate more complex primitives compared to other primitive-based methods. Then we compare the required number of parameters to represent 3D shapes with our representation and other methods. Finally, we demonstrate the power of our reconstruction method.
Table 2: Number of parameters. The average number of parameters for representing 3D shapes by different methods.

<table>
<thead>
<tr>
<th>Representation</th>
<th>SIF</th>
<th>OccNet</th>
<th>CvxNet</th>
<th>3DIAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of params</td>
<td>700</td>
<td>11M</td>
<td>7700</td>
<td>480</td>
</tr>
</tbody>
</table>

scheme to learn the semantic structures in an unsupervised manner. Moreover, we evaluate the effects of the designed constraints and the defined loss functions in reconstructing 3D shapes with high detailed appearances.

4.3.1 Complexity of primitives

We illustrate that our proposed constrained primitive is able to form more geometrical (e.g., curved) and topological (e.g., genus-one) complex shapes as shown in Figure 6. While the previous primitive-based methods such as cubes [40], ellipsoids [12], superquadrics [30], and convexes [10] cannot form such complex shapes.

4.3.2 The number of parameters

In section 3.1 we argue that a primitive may have no real solution when \( |p^{jk}(x, y, z)| < |p^k(x, y, z)| \) or \( |p^{jk}(x, y, z)| \) is non-negative for all points \((x, y, z) \in \mathbb{R}^3\) (i.e., no valid surface). Accordingly, our method can ignore some of the primitives among all \( M = 100 \) primitives by assigning positive definite coefficient matrices \( A \) to them and maintain a sufficient number of primitives. Therefore, these not-solvable primitives do not participate in reconstructing the surface \( S \). Note that, our method selects sets of primitives that are mainly different for inter-category shapes and have a large overlap for intra-category shapes. During the test phase, we can efficiently check the eigenvalues for the coefficient matrix of each primitive and eliminate the primitives with non-negative eigenvalues. Our experiments demonstrate that our network selects few primitives to reconstruct 3D shapes, as shown in Figure 5. In addition, since each quartic primitive can finally be identified with only 35 coefficients \( a_{ijk} \), the surface \( S \) of 3D shapes with 3DIAS representation can be represented with only \( 35 \times 13.71 \approx 480 \) number of parameters on average. 3DIAS requires 68.571\%, 0.004\%, and 6.234\% of the parameters used in SIF [12], OccNet [26], and CvxNet [10] on average to represent 3D shapes, respectively, see Table 2.

4.3.3 Unsupervised semantics segmentation

We also illustrate that our network learns a semantic structure without any part-level supervision such that one primitive usually covers the same part of reconstructed 3D shapes in the same class with 3DIAS representation. We evaluate the semantic structures on the PartNet[27] dataset having the labels of hierarchical parts of the shapeNet. The quantitative experiments in Figure 7 show that our method achieves better and comparable average accuracy compared to CvxNet [10] and BAE [7], respectively. In addition, 3DIAS achieves better accuracy than both methods for thin parts (e.g., arm). Moreover, our qualitative experiments illustrate that one primitive tends to cover the same semantic part as shown in Figure 8. This tendency is more pronounced for the dominant primitive that covers more points. For instance, the dominant primitives mainly cover the seat of chairs because most of the chairs have seat parts. Please see the supplementary material for more examples.
Table 3: Ablation study on constraints. We compare the effects of the center, the scale, and the closedness constraints in terms of IoU, Chamfer, and F-score. Note that in each configuration we ignore one or two constraints.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>IoU</th>
<th>Chamfer</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>-center</td>
<td>0.549</td>
<td>0.387</td>
<td>46.72</td>
</tr>
<tr>
<td>-scale</td>
<td>0.559</td>
<td>0.261</td>
<td>48.45</td>
</tr>
<tr>
<td>-scale, -closedness</td>
<td>0.546</td>
<td>0.280</td>
<td>44.44</td>
</tr>
<tr>
<td>All</td>
<td>0.575</td>
<td>0.197</td>
<td>52.22</td>
</tr>
</tbody>
</table>

4.3.4 Effects of constraints

We study the effect of our designed constraints on reconstructing 3D shapes. In each experiment, we evaluate or baseline by ignoring one or more constraints. The quantitative results based on volumetric IoU, Chamfer, and F-Score show the importance of each constraint, see Table 3. We believe these constraints encourage the network to reconstruct a detailed 3D shape, especially the center constraint.

4.3.5 Effects of losses

While $L^{\text{sign}}$ for the inside/outside points tries to distinguish inside and outside of 3D shapes, it is not enough to achieve a detailed surface due to the lack of sample points near the surface. Therefore, points on the surface and their normal vectors can facilitate the reconstruction. Note that normal vectors carry important information on 3D geometry, such as the local orientation of surfaces. Accordingly, we use $L^{\text{sign}}$ loss and $L^n$ loss for the points on the surface to better approximate the surfaces. We evaluate the effects of each $L^{\text{sign}}$, $L^n$, and their combination by excluding them for the points on the surface and summarize the results in Table 4. Note that for all the experiments in Table 4 we do not exclude $L^{\text{sign}}$ for the inside/outside points. The results indicate the importance of points on the surface to achieve more detailed 3D shapes.

5. Conclusion

In this paper, we propose a primitive-based representation and a learning scheme in which the primitives are learnable implicit algebraic surfaces that can jointly approximate 3D shapes. We design various constraints and loss functions to achieve high-quality and detailed 3D shapes. We experimentally demonstrate that our method outperforms state-of-the-art methods in most of the metrics. Moreover, we illustrate that our method can learn semantic meanings without part-level supervision by automatically selecting sets of primitives parametrized by only a few parameters. In the future, we will utilize the solvability of the designed primitives to develop a soft renderer which leads to reconstruct the 3D shapes with self-supervised learning.

Acknowledgement

This work was supported in part by an IITP grant funded by the Korean government [No. 2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)].
References


[38] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. pages 2107–2115, 10 2017. 2

